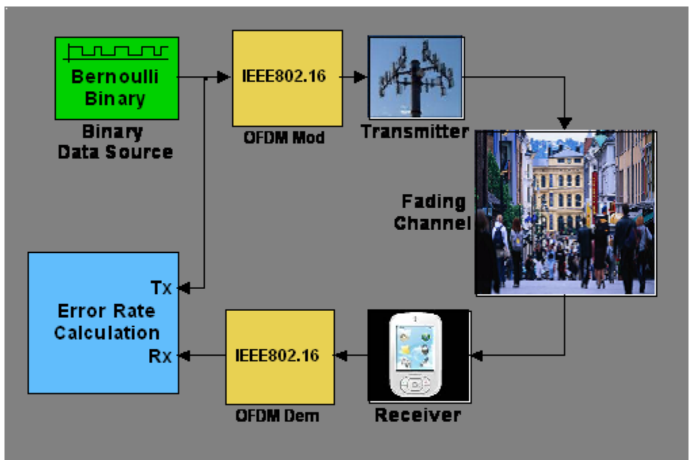
Wireless Communications with Matlab and Simulink: IEEE802.16 (WiMax) Physical Layer



by

Roberto Cristi

Professor
Dept of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943
August 2009

Table of Content:

Introduction to IEEE 802.16 and WiMax

Existing “Area Networks” for wireless communications:

- **Personal** (PAN), up to a few meters. It requires simple “thumb like” transmitters receivers. Typical: Bluetooth.
- **Local** (LAN), up to 300m. It requires simple “box like” devices. Typical: WiFi (IEEE802.11)
- **Wide** (WAN), up to a few miles. It requires towers and cellular technology. Typical: W-CDMA, CDMA 2000, UMTS …

IEEE 802.16 (WiMax) are possible future technologies for WAN.

IEEE 802.16 and WiMax

• IEEE802.16 is a standard for Broadband Wireless Access (BWA) Air Interface. It is purely technical (not commercial);

•“The **WiMAX Forum®** is an industry-led, not-for-profit organization formed to certify and promote the compatibility and interoperability of broadband wireless products based upon the harmonized IEEE 802.16/ETSI HiperMAN standard. A WiMAX Forum goal is to accelerate the introduction of these systems into the marketplace. WiMAX Forum Certified™ products are fully interoperable and support broadband fixed, portable and mobile services. Along these lines, the WiMAX Forum works closely with service providers and regulators to ensure that WiMAX Forum Certified systems meet customer and government requirements” (from the WiMax website www.wimaxforum.org)

Current WAN Technologies for Voice and Data

3G: CDMA2000 and UMTS. All based on Spread Spectrum

3.5G: increased capacity by combining CDMA with TDM (Time Division Multiplexing). Voice and Data on separate channels. Current technology;

4G : to provide voice, data, multimedia services at low cost on an all IP (packets) network. IEEE802.16 (WiMax) is one of the technologies considered.

TODAY: voice and data on separate networks;

TOMORROW: voice and data on the same network. Data using Voice Over IP (VoIP). Advantages: flexibility, control of QoS, scalable.

Evolution of IEEE802.16:

802.16	Dec 2001	10-66GHz	Fixed, LOS	Single Carrier
802.16-2004	June 2004	2-11GHz	Fixed, NOLOS	SC, 256 OFDM, 2048 OFDMA
802.16e-2005	Dec 2005	2-6GHz	Fixed, Mobile, NOLOS	SC, 256OFDM, 128, 512, 1024, 2048 OFDMA

In addition, IEEE802.16-2004 and 2005 have options based on Multi Antennas techniques.

Introduction to Simulink

- Matlab based
- Both Continuous Time and Discrete Time Simulation
- Based on Blocksets
- Model Based Design: a software model of the environment can be developed and the design can be tested by simulation
- Transition between “ideal” algorithms (infinite precision, floating point) to “real world” algorithms (finite precision, fixed point);
- Automatic Code Generation: once the design is tested and validated, real time code can be automatically generated for the target platform
- Continuous Test and Verification

Advantages (from the MathWorks slide)

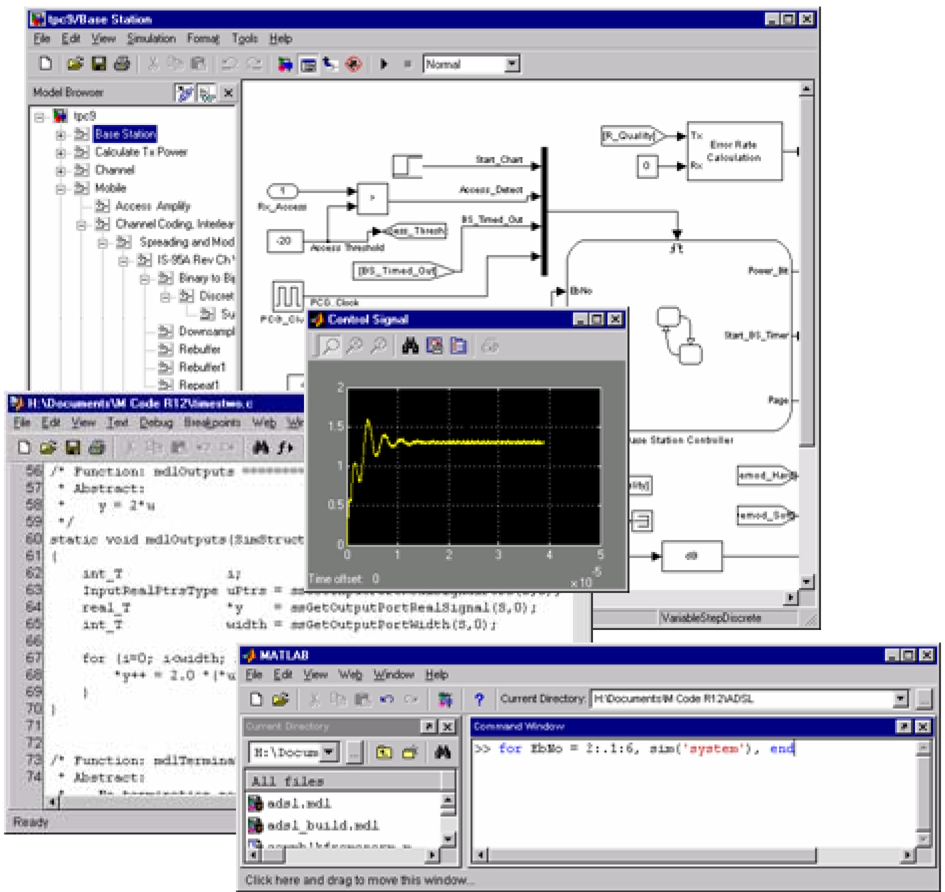
Innovation
- Rapid design iterations
- “What-if” studies
- Unique features and differentiators

Quality
- Reduce design errors
- Minimize hand coding errors
- Unambiguous communication internally and externally

Cost
- Reduce expensive physical prototypes
- Reduce re-work
- Reduce testing

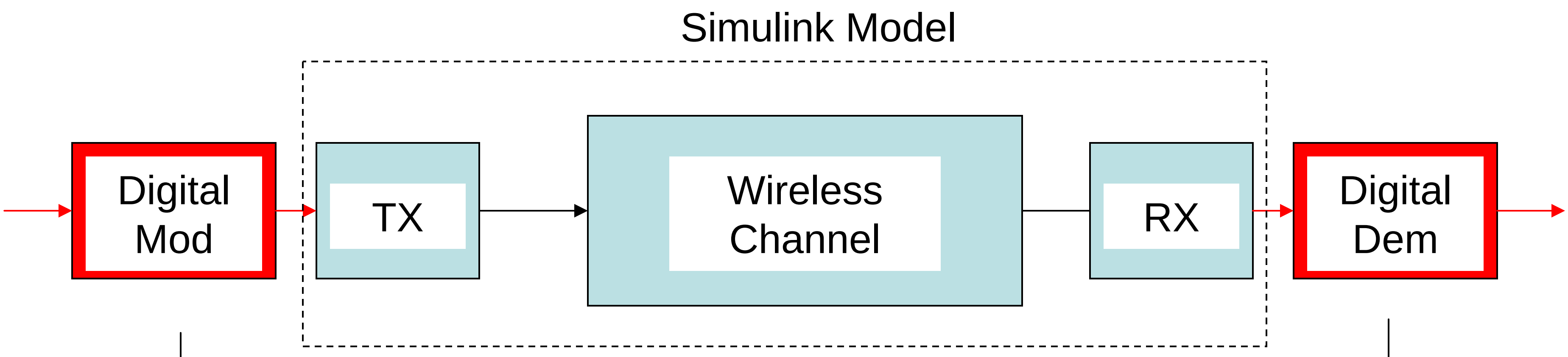
Time-to-market
- Get it right the first time

Simulink



- Hierarchical block diagram design and simulation tool
  - Built-in notions of time and concurrency
- Digital, analog/mixed signal and event driven
- Visualize Signals
- Co-develop with C code
- Integrated with MATLAB

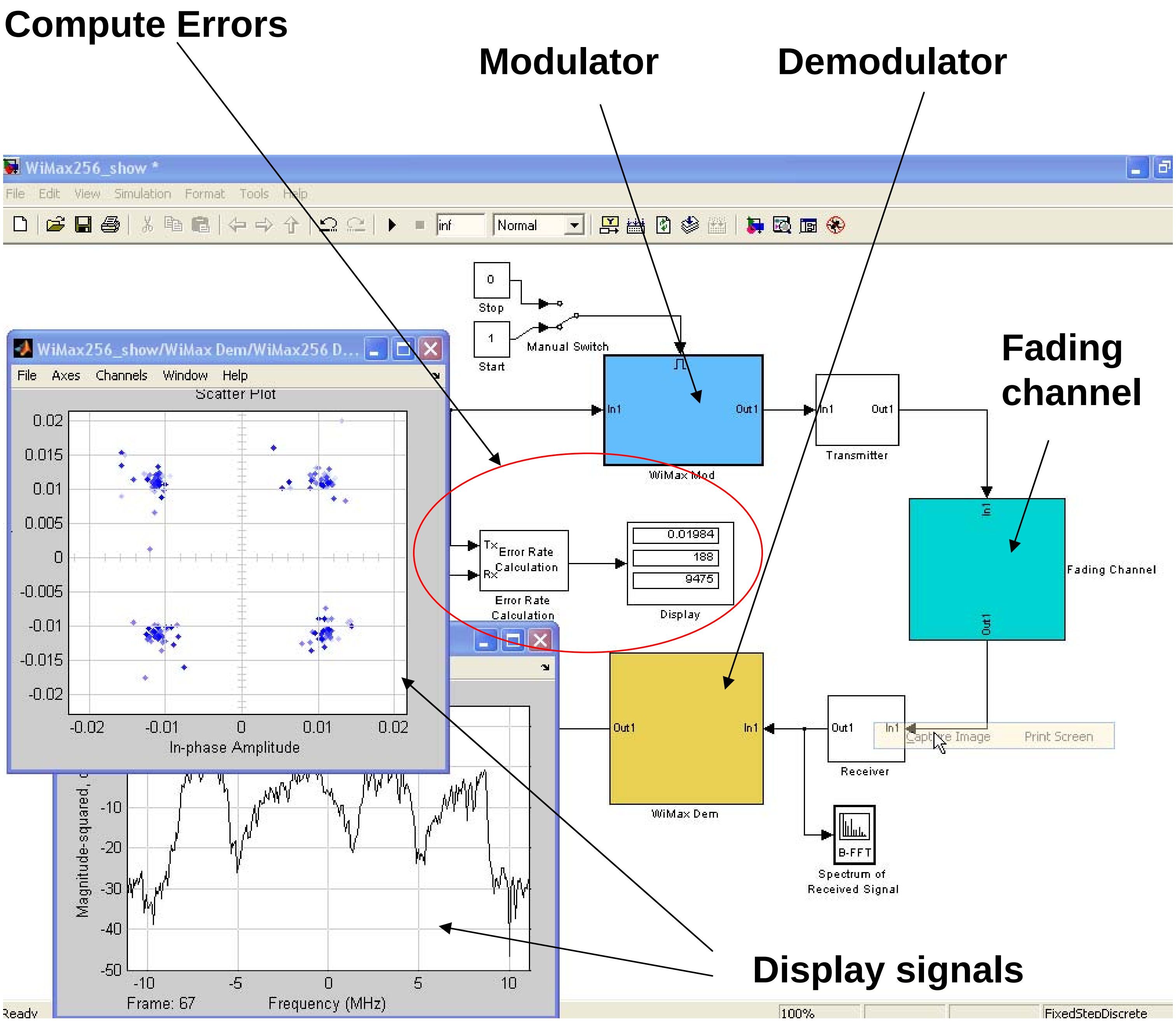
Example



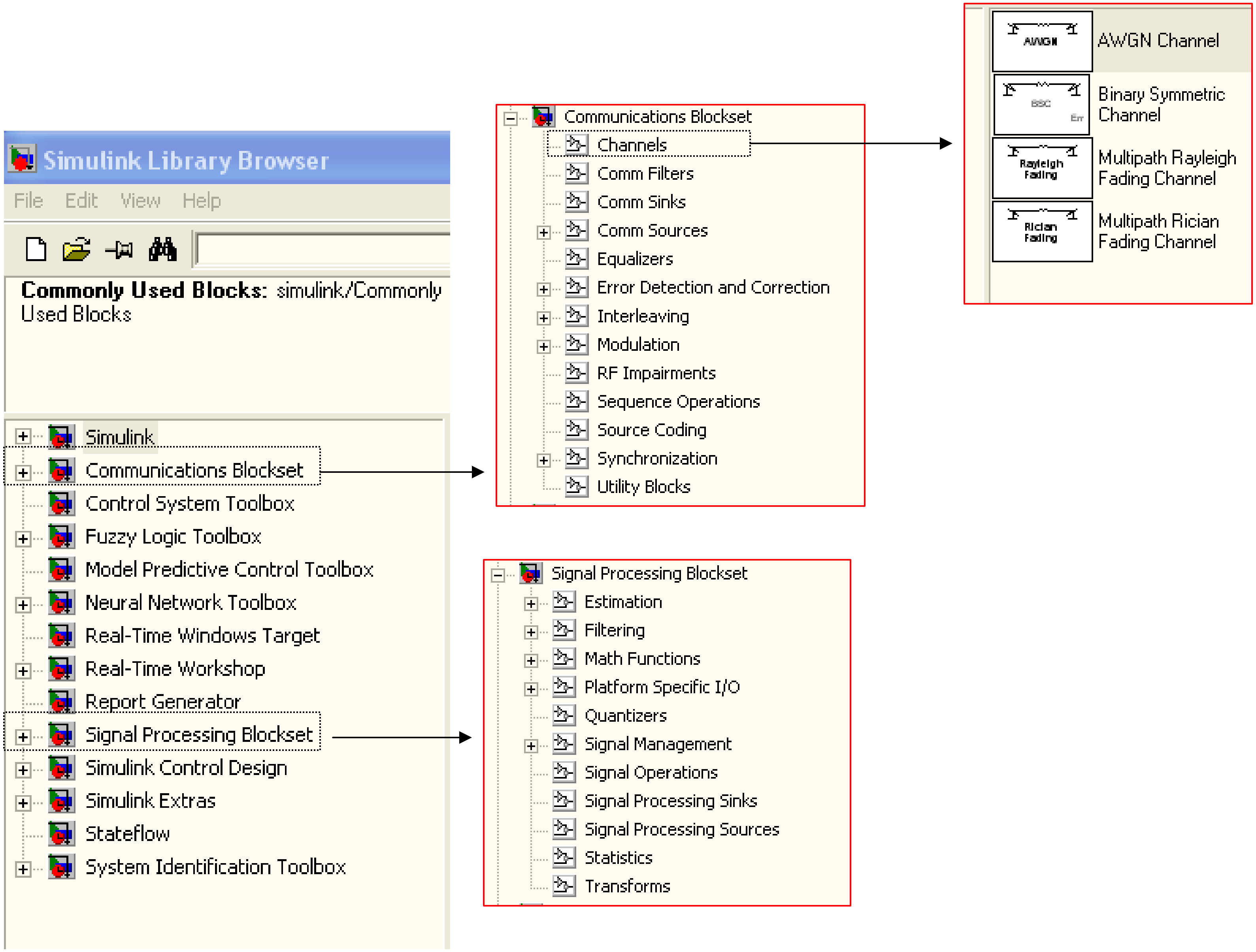
- Design
- Test
- Automatically Generate Code

… 0001010101110101010100101111

0010101010010010101001001001….


Compute Errors
Modulator
Demodulator
Fading channel
Display signals
WiMax256_show *
File Edit View Simulation Format Tools Help
Stop
Start
Manual Switch
WiMax Mod
Transmitter
Fading Channel
Error Rate Calculation
Display
0.01984
188
9475
WiMax Dem
Receiver
Spectrum of Received Signal
WiMax256_show/WiMax Dem/WiMax256 D...
File Axes Channels Window Help
Scatter Plot
In-phase Amplitude
Frame: 67
Frequency (MHz)
Ready
FixedStepDiscrete

Simulink has a very rich library of blocksets:



1. Introduction to Digital Signal Processing and Matlab

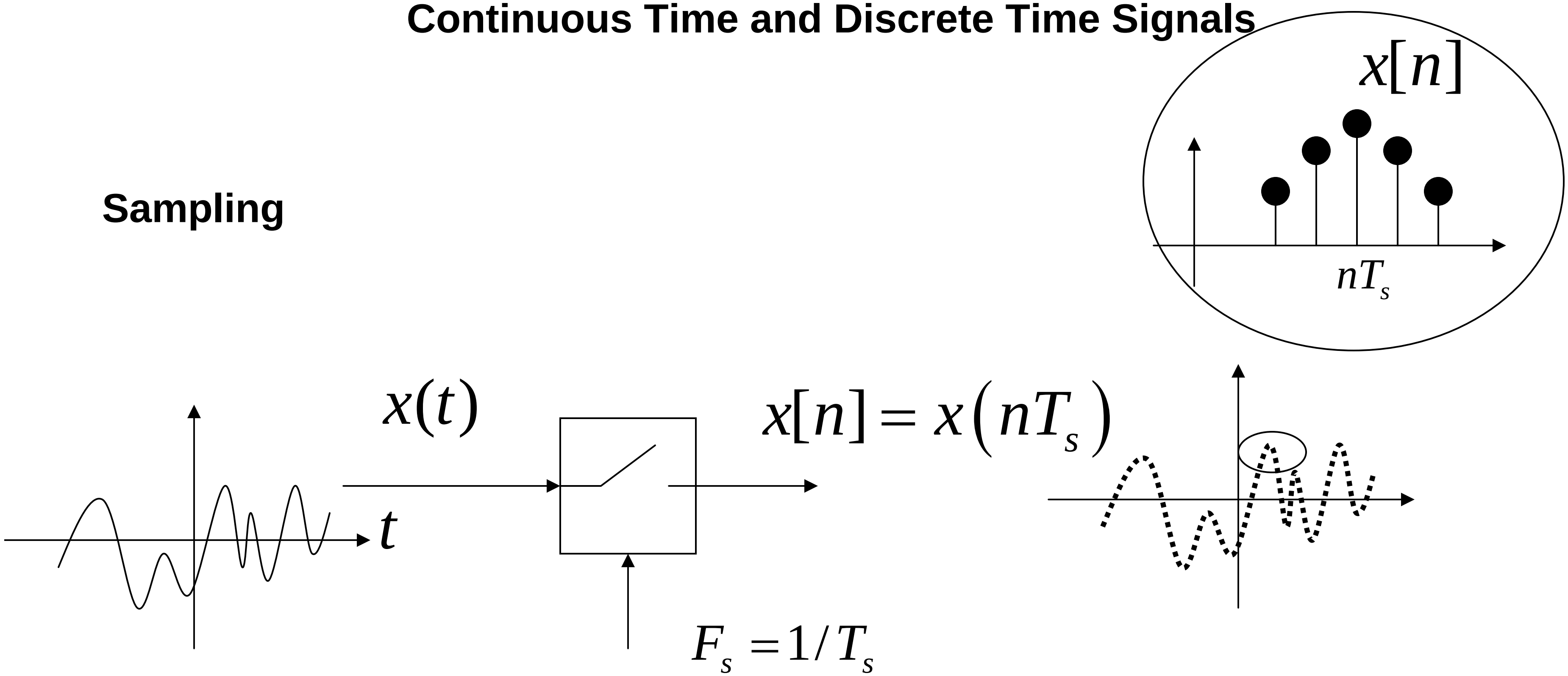
1. Discrete time Signals and Systems
2. Fast Fourier Transform (FFT) and its Inverse (IFFT)
3. Convolution and Correlation

1. Discrete Time Signals

1. Continuous Time to Discrete Time
2. Fundamental Signals: Delta Function, Sinusoid, Complex Exponential

Continuous Time and Discrete Time Signals

Sampling

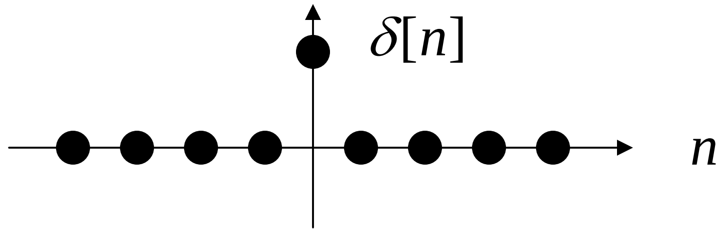


F_s **Sampling frequency (Hz=1/sec)**

T_s **Sampling interval (sec)**

Fundamental Discrete Time Signals

1. “Delta” or “Impulse”



2. Sinusoid

phase

$$x[n] = A\cos\left(2\pi F_0 t + \alpha\right)\Big|_{t=nT_s} = A\cos\left(\omega_0 n + \alpha\right)$$

amplitude

$$\boxed{\omega_0 = 2\pi \frac{F_0}{F_s}}$$

Frequency (Hz)

Digital Frequency (radians)

Example: $F_s = 10kHz$

$$x[n] = 10\cos\left(4,000\pi t + 0.1\right)\Big|_{t=nT_s} = 10\cos\left(\omega_0 n + 0.1\right)$$

Digital frequency:

$$\begin{matrix} F_0 = 2kHz \\ F_s = 10kHz \end{matrix} \Rightarrow \omega_0 = 2\pi\frac{2,000Hz}{10,000Hz} = \frac{2\pi}{5}rad$$

Generate it in matlab: **plot_a_sinusoid.m**

n=0:100; ← n=[0,1,2,…,100]

A=10;

w0=2*pi/5;

alpha=0.1;

x=A*cos(w0*n+alpha); ← x=[10*cos(0.1), … , 10*cos((2*pi/5)*100+0.1)]

plot(x);

3. Complex Exponentials

$$y[n] = Ae^{j(\omega_0 n + \alpha)} = \left(Ae^{j\alpha}\right)e^{j\omega_0 n}$$

$$= \underbrace{A\cos(\omega_0 n + \alpha)}_{\text{Real Part}} + j\underbrace{A\sin(\omega_0 n + \alpha)}_{\text{Imaginary Part}}$$

where $j = \sqrt{-1}$ imaginary basis

Then a sinusoid becomes the “real part” of a complex signal

$$\boxed{A\cos(\omega_0 n + \alpha) = \mathrm{Re}\left\{Ae^{j(\omega_0 n + \alpha)}\right\}}$$

Sinusoids and Frequency Spectrum

Complex exponentials are the “building blocks” of signals and systems.

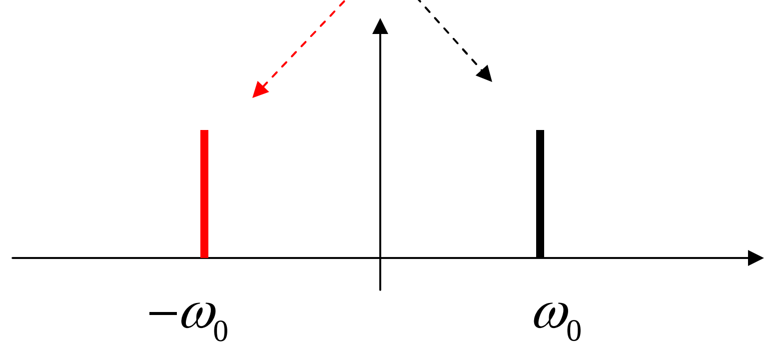
Reasons: all operations of interest boil down to multiplications or divisions

A sinusoid in terms of complex exponentials can also be written as

$$\boxed{A\cos(\omega_0 n+\alpha)=\frac{A}{2}e^{j(\omega_0 n+\alpha)}+\frac{A}{2}e^{-j(\omega_0 n+\alpha)}}$$

with two frequencies:

positive and negative

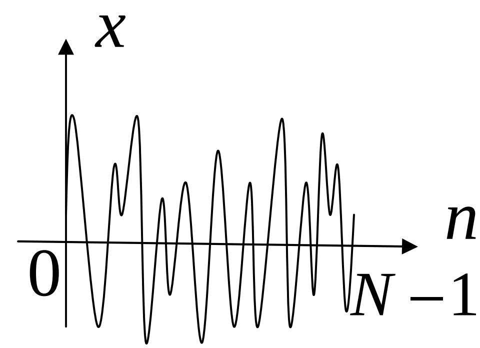


The Fast Fourier Transform (FFT) and its Inverse (IFFT)

1. Definitions
2. Examples
3. Computation in Matlab
4. Symmetry

The Fast Fourier Transform (FFT)

Given a finite sequence of data



$$x[n],\ n = 0,\dots,N-1$$

1. Define the FFT

$$\boxed{X = FFT\{x\}}$$

where

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk\frac{2\pi}{N}n} \qquad k = 0,\dots,N-1$$

2. … and the IFFT (Inverse FFT)

$$\boxed{x = IFFT\{X\}}$$

where

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{jk\frac{2\pi}{N}n} \qquad n = 0,\dots,N-1$$

Meaning:

$$X[k],\, k = 0, \ldots, N-1$$

is the component of the spectrum due to frequency

$$F = k\frac{F_s}{N}$$

F_s ← sampling frequency

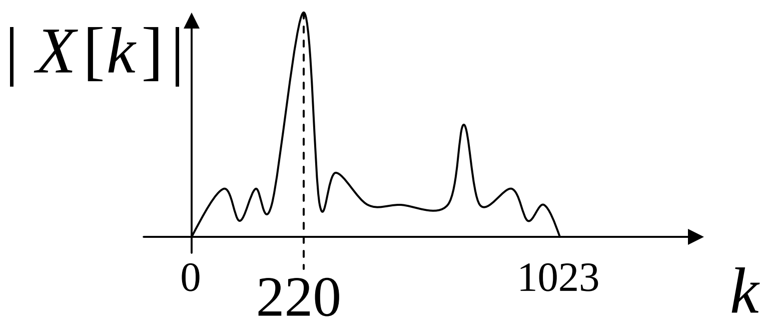
N ← data length

Example:

Let

$$F_s = 10kHz$$

$$N = 1024$$



$$\boxed{F = 220\frac{10}{1024}kHz = 2.15kHz}$$

Example: **example_of_fft.m**

Data length N=1024;

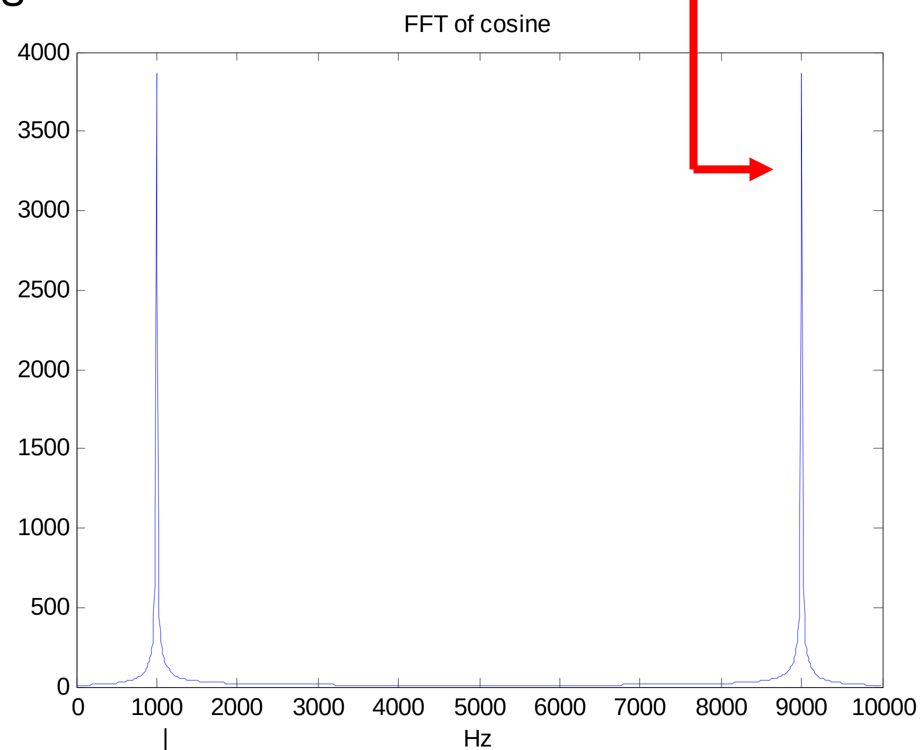
Sampling Frequency Fs=10kHz

X=fft(x);

k=0:1023; f=k*Fs/N; % frequency axis

plot(f,abs(X))

This corresponds to the "negative frequency" (disregard)



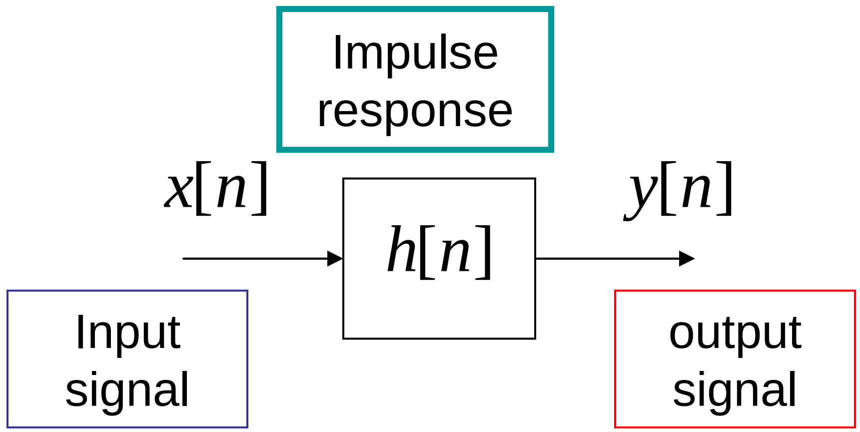
Freq. = 1kHz

Convolution and Correlation

1. Convolution as system response
2. Autocorrelation of data
3. Crosscorrelation between data sets
4. Estimation of Impulse Response using Crosscorrelation

Operations of Interest

1. Convolution. To compute the output of a Linear Time Invariant system

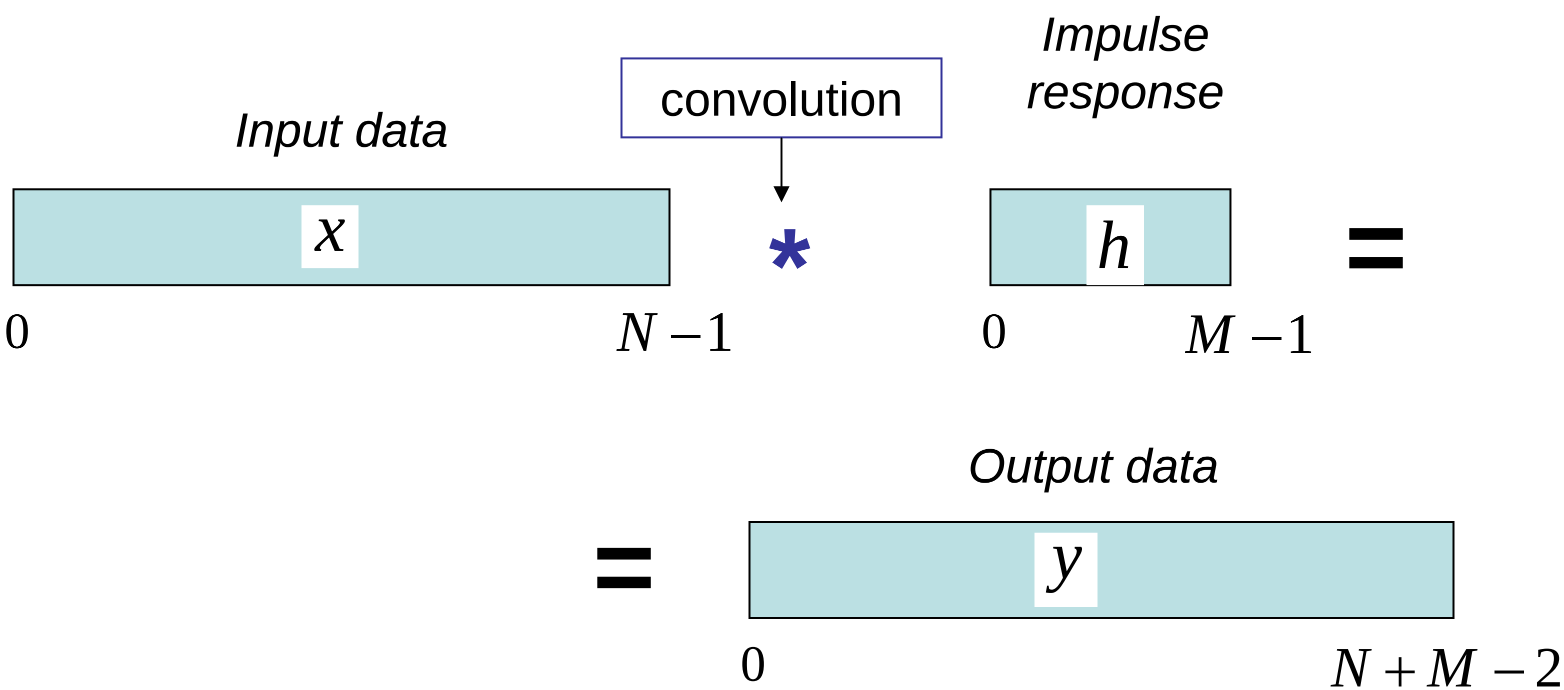


Definition of Convolution:

$$y[n] = h[n] * x[n] = \sum_{\ell=-\infty}^{+\infty} h[\ell]x[n-\ell]$$

In general you deal with sequences of finite length

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \ldots + h[M-1]x[n-M+1]$$



In matlab:

Let

1. *h* be the vector of impulse response;

2. *x* be the input vector

Then the output vector

y=conv(*h*,*x*);

Example: **convolution_of_finite_sequences.m**

```
h=[1,0,0,0.5,0,-0.2,0,0.1];            % impulse response
n=0:200;   x=2*cos(0.1*pi*n);          % input signal
y=conv(h,x);                           % output signal
plot(y)
```

2. Auto Correlation.

For a signal with zero mean, to see if the samples are “correlated” with each other

Definition of Auto Correlation: $$r_x[m] = \sum_{n=-\infty}^{+\infty} x[n]x^*[n-m]$$

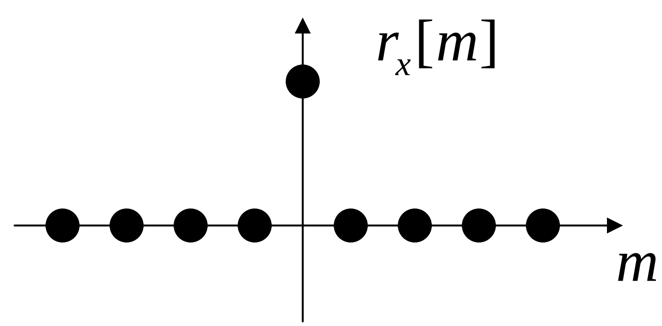
Again, you deal with signals of finite length

$$x[n],\ n = 0,\ldots, N-1$$

$$\boxed{r_x[m] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x^*[n-m], \quad m = -N+1,\ldots, N-1}$$

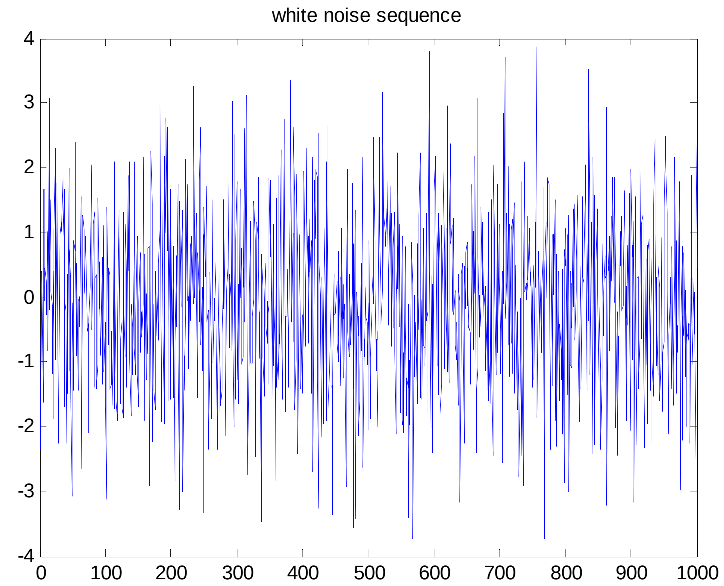
Example: White Noise with standard deviation σ_x

$$r_x[m] = \begin{cases} \sigma_x^2 = \frac{1}{N}\sum_{n=0}^{N-1} |x[n]|^2, & \text{if } m = 0 \\ \approx 0, & \text{otherwise} \end{cases}$$

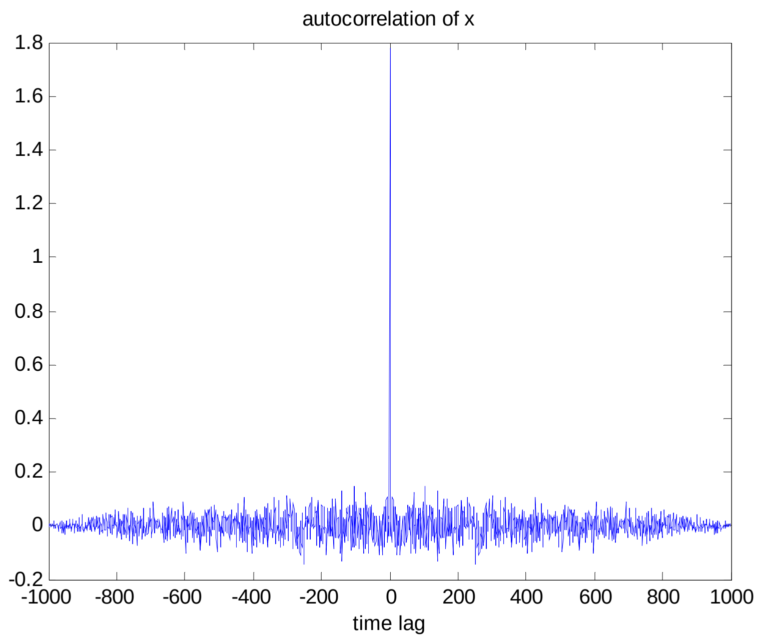


Example: **xcorr_of_white_noise.m**

```
% data
sigx=sqrt(2)
x=sigx*randn(1,1000);
plot(x);
```




```
% autocorrelation
N=length(x);
rx=xcorr(x)/N;
max_lag=length(x)-1;
m=-max_lag:max_lag;
plot(m,rx)
```



2. Cross Correlation.

To see if two signals are "correlated" with each other

$$r_{yx}[m] = \sum_{n=-\infty}^{+\infty} y[n]x^*[n-m]$$

With finite length signals:

$$x[n],\ n = 0,\ldots, N-1$$
$$y[n],\ n = 0,\ldots, M-1$$

$$\boxed{r_{yx}[m] = \frac{1}{M}\sum_{n=0}^{M-1} y[n]x^*[n-m],\ m = -\max(N, M)+1,\ldots, \max(N, M)-1}$$

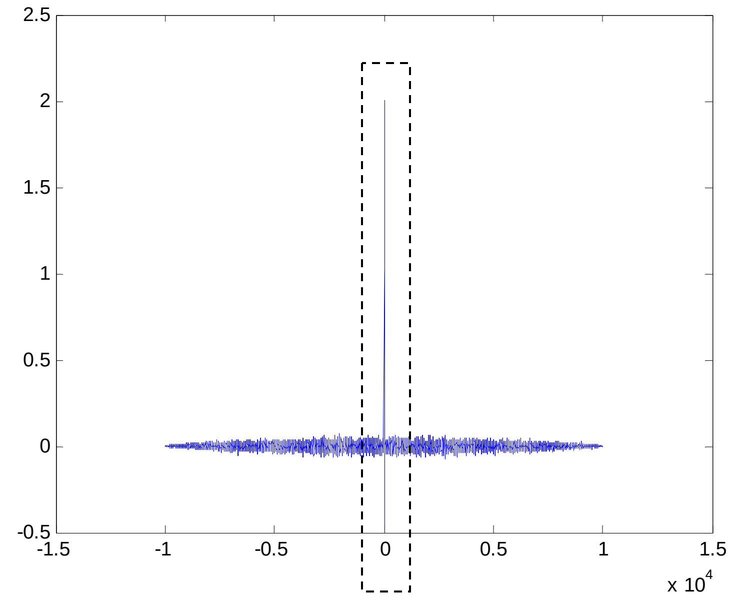
Case of Interest: estimation of the impulse response of a Linear Time Invariant system from input-output data

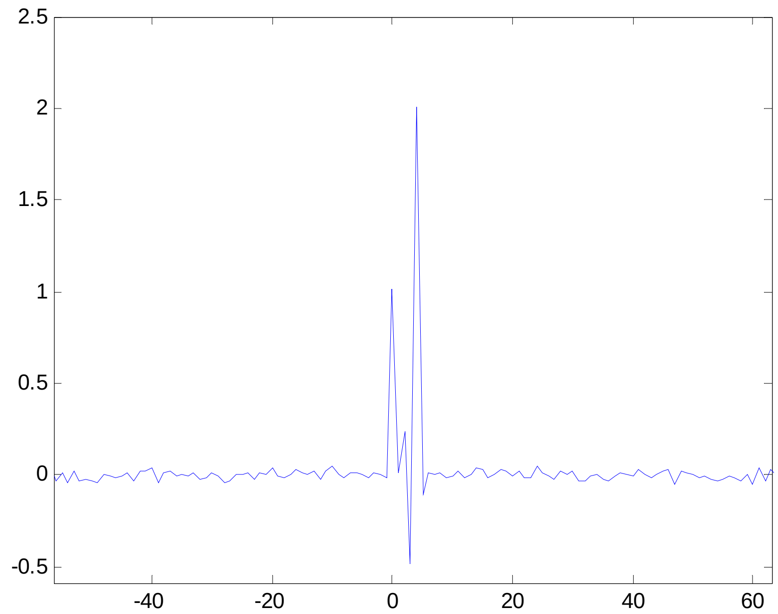
$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

If $x[n]$ is white noise, then $h[n] = r_{yx}[n] / r_x[0]$

Example: **impulse_response_with_xcorr.m**

```
h=[1, 0, 0.2, -0.5, 2, -0.1]; % impulse response
y=conv(h,x);
ryx=xcorr(y,x) /length(y)
h_est=ryx/ryx(length(x));
```





zoom in

Lab 1: Introduction to DSP and Matlab

A. Sinusoids and the FFT:

1. Generate a sinusoidal signal of a given frequency
2. Check its frequency spectrum

B. White Noise, Convolution, Correlation:

1. Generate a white noise signal with a given covariance
2. Determine the output of a given Linear Time Invariant System
3. From the input-output data, estimate the impulse response of the system

A. Sinusoids and the FFT:

A.1 Generate a sinusoid with the following parameters and plot it vs time:

Amplitude	$A = 5.0$
Frequency	$F_0 = 5.0\ kHz$
Sampling Frequency	$F_S = 15.0\ kHz$
Phase	$\alpha = 30^0$
Length	$128\ \text{samples}$

Reference: **plot_a_sinusoid.m**

A.2 Take the FFT of the sinusoid you generated, plot its magnitude (absolute value), and verify that you obtain the frequency you expect.

Reference: **example_of_fft.m**

B. White Noise, Convolution, Correlation:

1. Generate a gaussian white noise signal with the following parameters:

Standard Deviation	$\sigma_x = 5$
Length	$N = 10{,}000$ data points

Plot its autocorrelation and verify that it is as expected.

2. This signal is the input to a system with impulse response

$$h = [1, 0, -2, 0.5, 0, 0, 0, 0.3]$$

Determine the output sequence and verify that the crosscorrelation between input and output is a good estimate of the impulse response of the system.

Reference: **impulse_response_with_xcorr.m**

2. Digital Communications Fundamentals and the Additive White Gaussian Noise (AWGN) Channel

1. *General Structure of a Digital Communication System*
2. *Channel Losses and Noise*
3. *Complex Baseband Representation*
4. *Bit Error Probabilities*
5. *Simulink Implementation*

References:

J. Proakis, M. Salehi, "Digital Communications," Prentice Hall, 2007

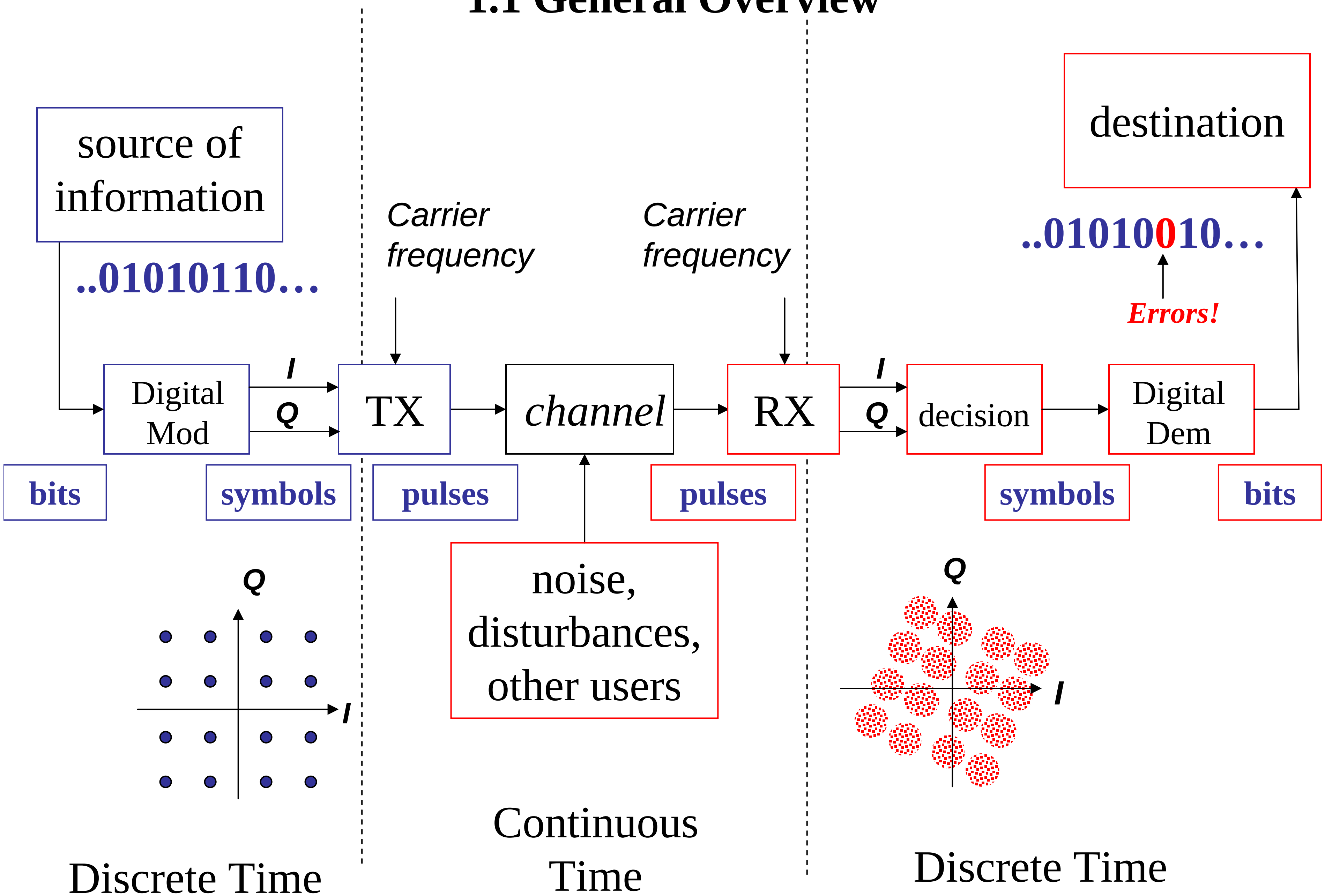
1. General Structure of a Digital Communications System

1.1 General Overview

1.2 Goals

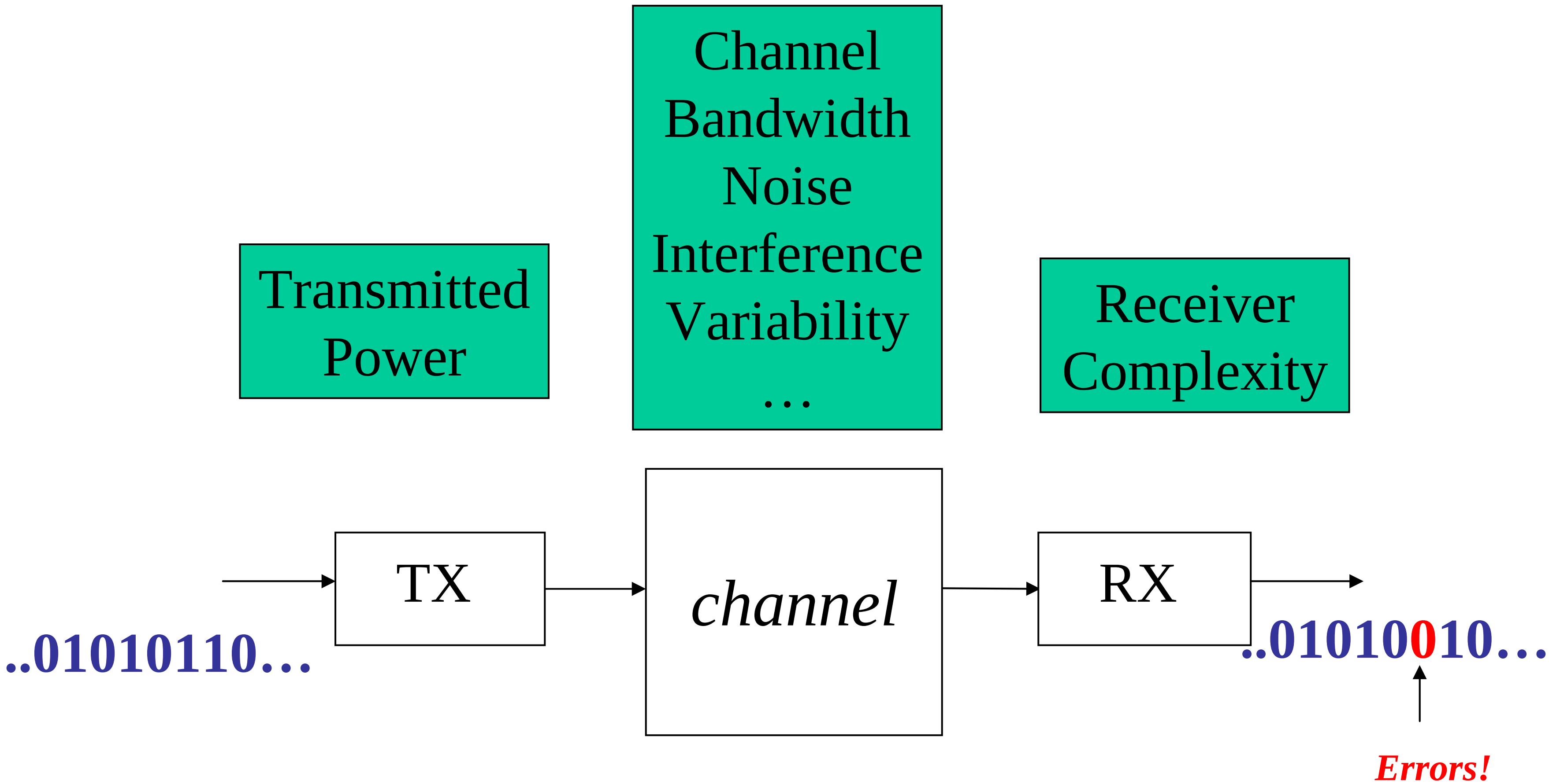
1.3 Parameters of Interest

1.1 General Overview

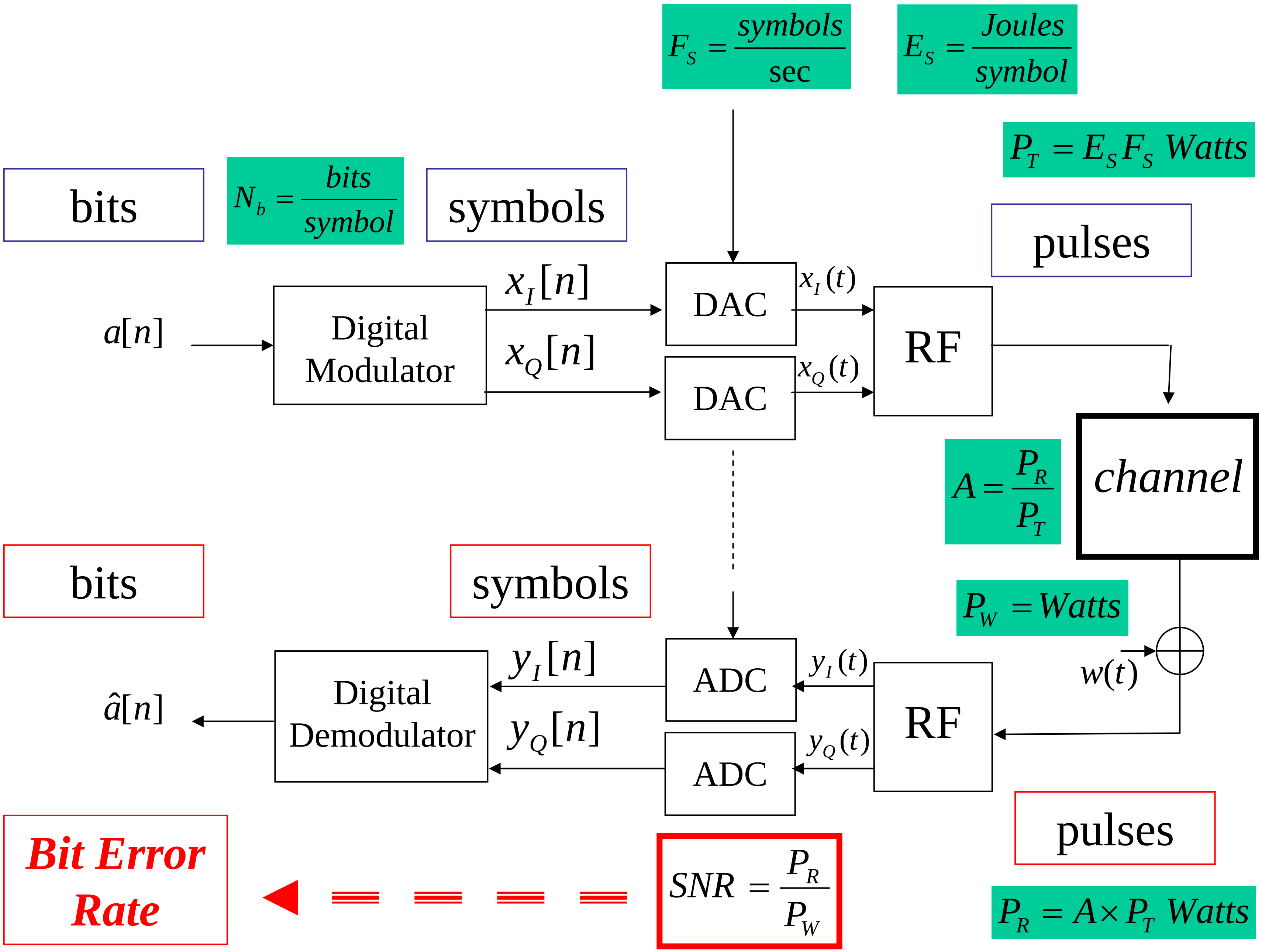


1.2 Goals

- Bit Error Rate within acceptable values;
- Stay within the available resources:



1.3 Parameters



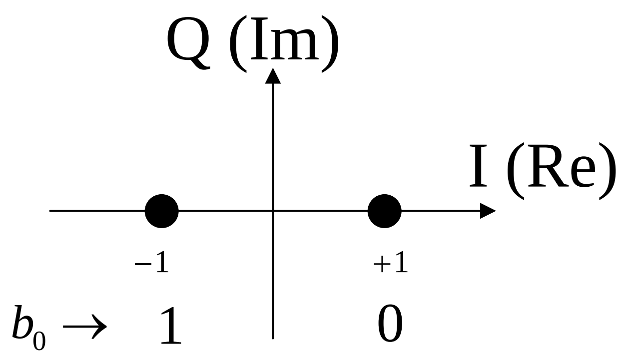
Bits to Symbols:

the Constellation Mapping in 802.16 (Gray Code)

BPSK

$[b_0]$

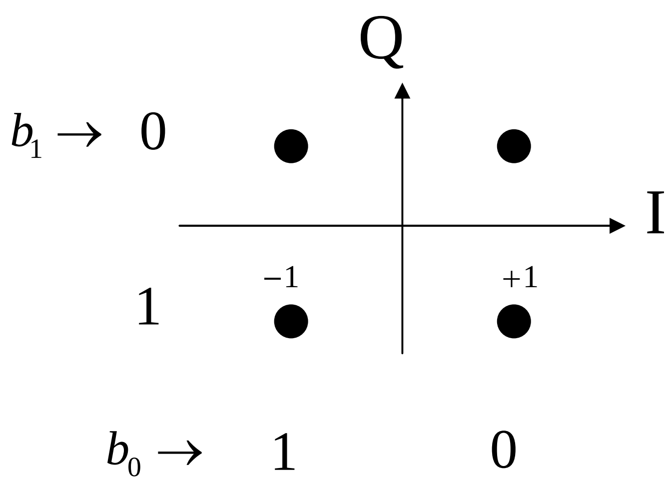
$N_b = 1$ bit/symbol



4-QAM
(QPSK)

$[b_0, b_1]$

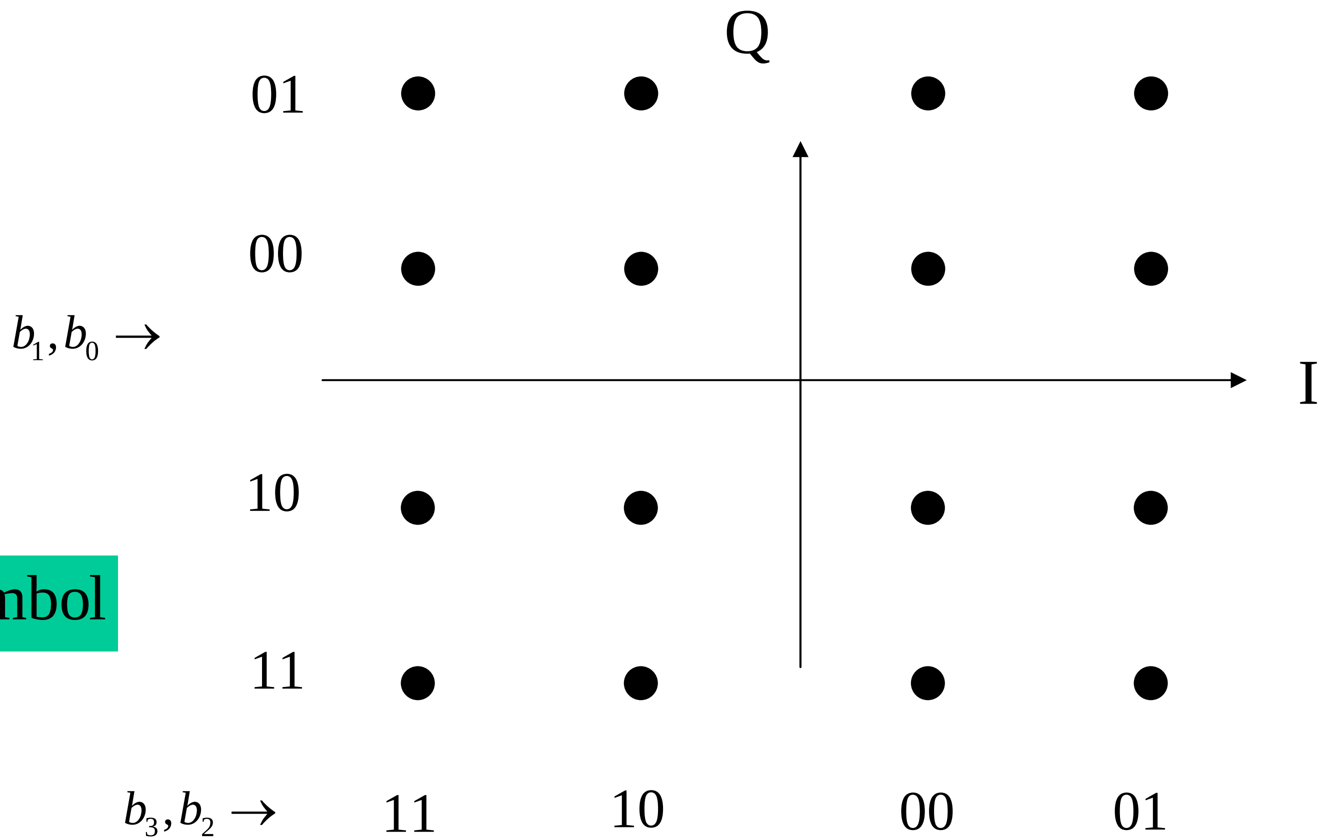
$N_b = 2$ bits/symbol



16-QAM

$[b_0, b_1, b_2, b_3]$

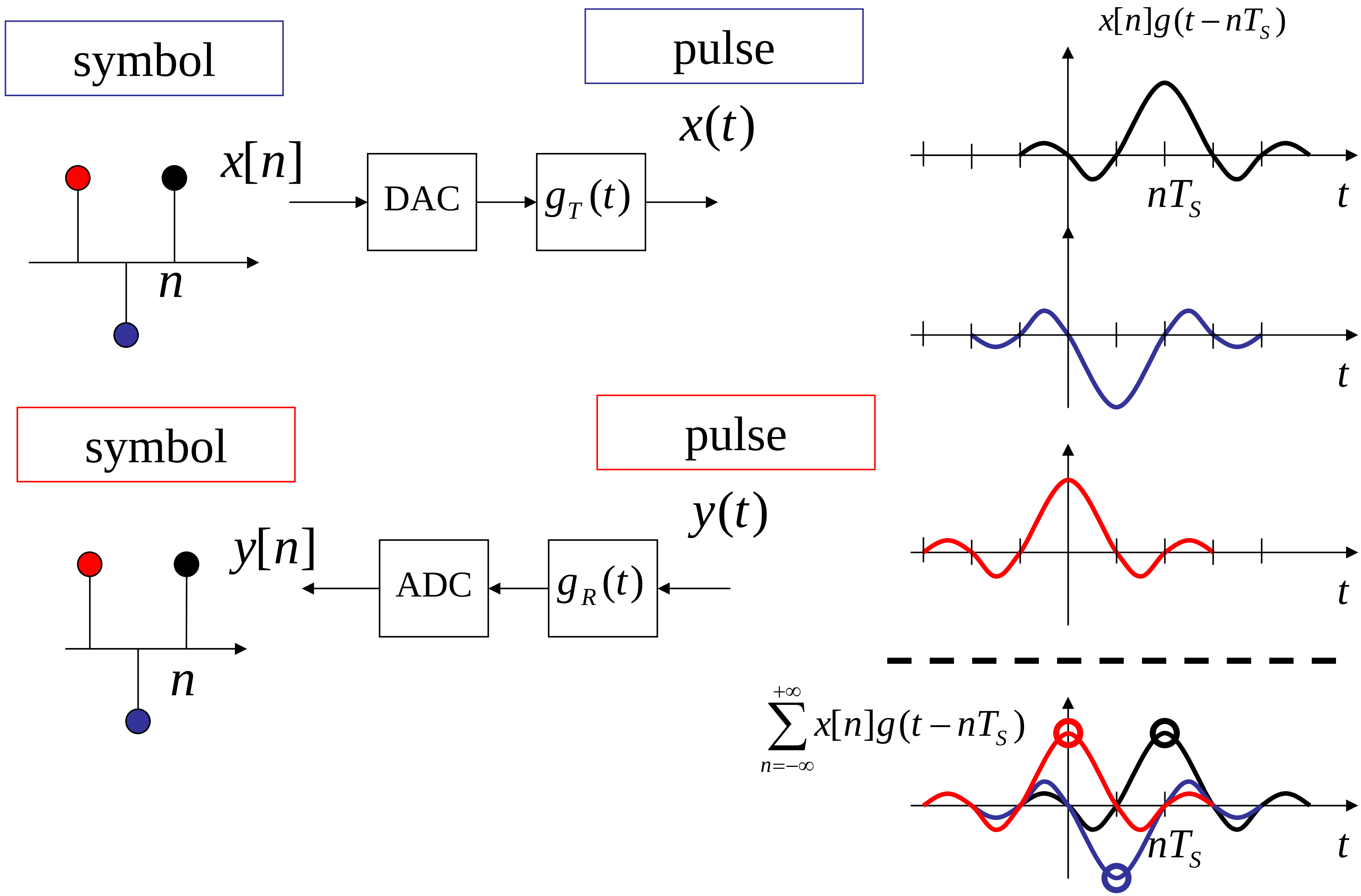
$N_b = 4$ bits/symbol



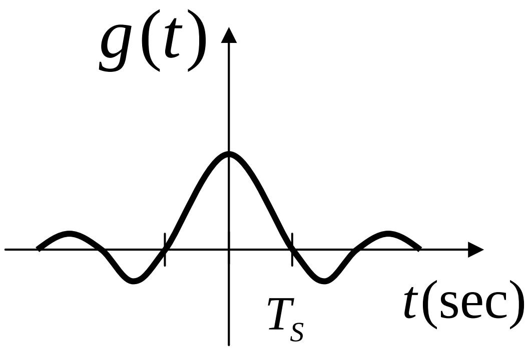
<u>Notice</u> : only one bit difference between close neighbors

Symbols to Pulses:

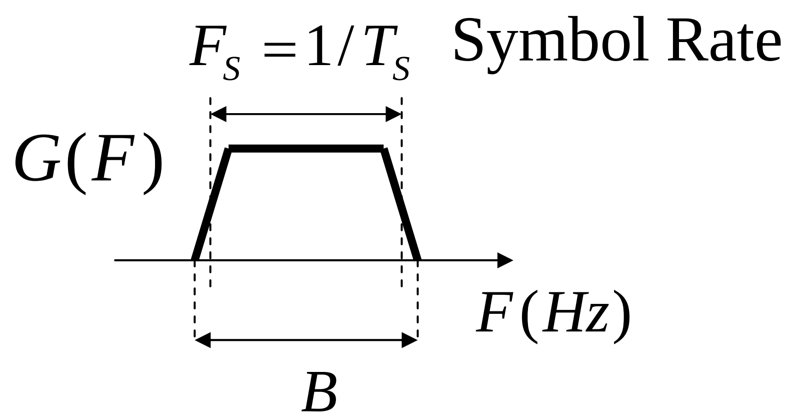
Digital to Analog (DAC) and Analog to Digital (ADC) Conversion



Bandwidth, Symbol Rate and Transmitted Power



Symbol interval

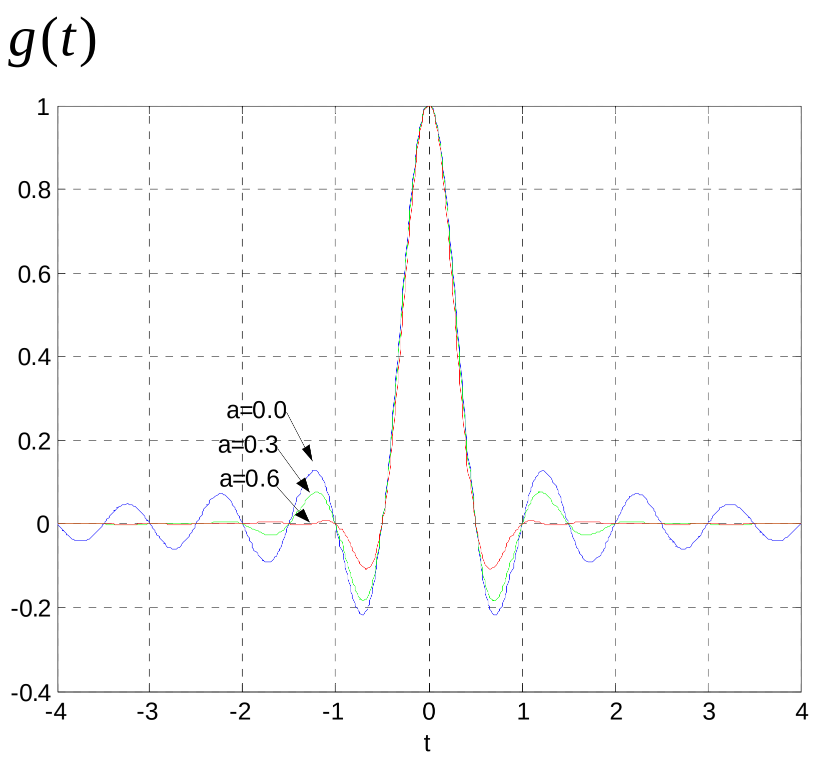


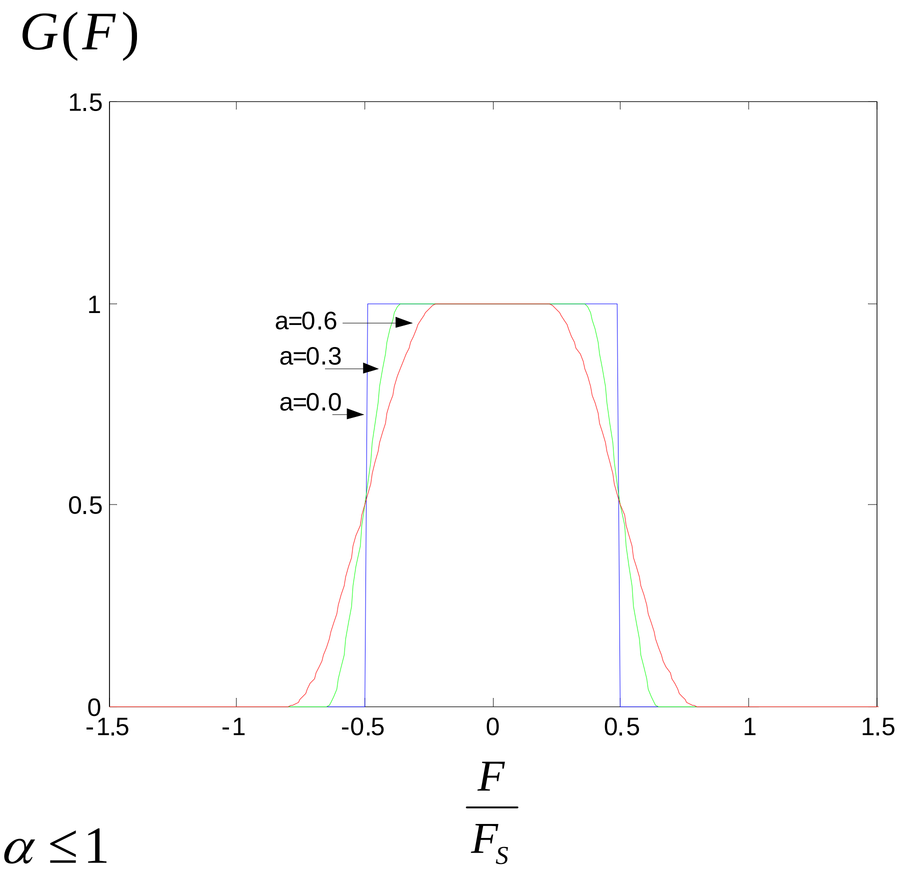
Bandwidth

Energy per symbol $$E_S = \int_{-\infty}^{+\infty} |g(t)|^2 \, dt = \int_{-\infty}^{+\infty} |G(F)|^2 \, dF$$

Transmitted Power $$P_T = \frac{E_S}{T_S} = E_S \times F_S$$

Typical: Raised Cosine





$$0 \le \frac{B}{F_S} - 1 = \alpha \le 1$$

- smaller bandwidth
- more ISI

- larger bandwidth
- less ISI

2. Channel Losses and Noise

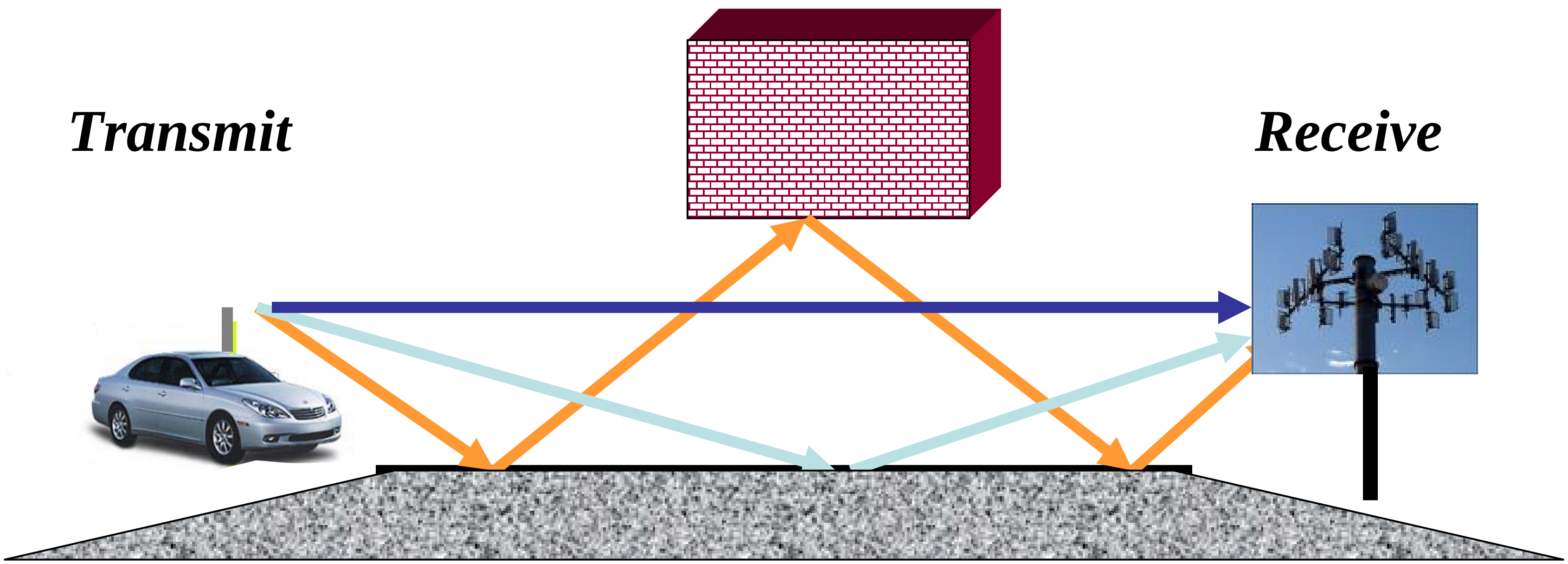
2.1 Transmitted Pulses to Received Pulses

2.2 Energy per Symbol, Power Spectral Density

2.3 Signal to Noise Ratio and "Eb/N0"

2.1 Transmitted Pulses to Received Pulses:

- **attenuation:** free space, obstacles, foliage …
- **noise:** thermal, interferences from other systems/users
- **multipath:** reflections from buildings, structures, hills …
- **doppler shift:** motion of transmitter, receivers, reflectors …



Channel Losses and Noise

Transmitted Power

$P_T = E_S F_S \; Watts$

Channel Attenuation

channel

$A = \dfrac{P_R}{P_T}$

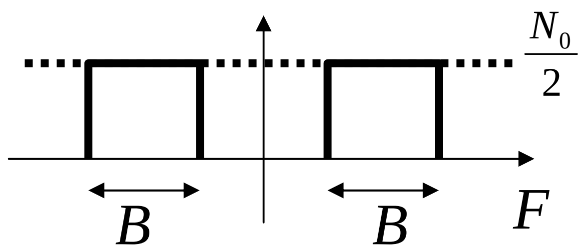
Received Power

$P_R = A \times P_T \; Watts$

$w(t)$

Noise

Noise Power Spectral Density



$P_W = N_0 \times B \; Watts$

2.2 Energy per Symbol and Power Spectral Density

Energy per symbol

Symbol Rate

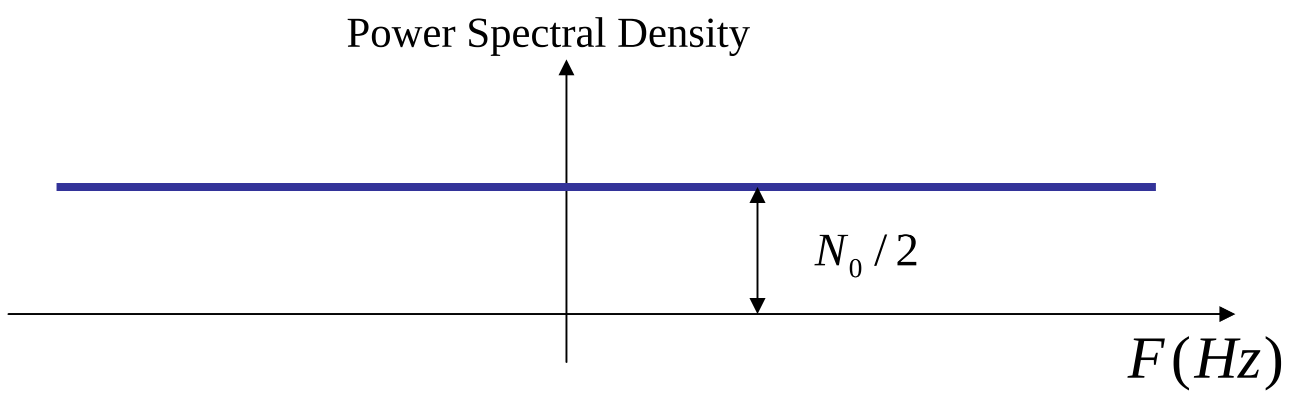
$$SNR = \frac{\text{Signal Power}}{\text{Noise Power}} = \frac{E_S \times F_S}{N_0 \times B}$$

Bandwidth

Noise Power Spectral Density

Thermal Noise

- Present in all electronic systems, it is dependent on temperature;
- It is "white", in the sense that it is equally spread in frequency.



$$\boxed{N_0 = kT = (1.383\times 10^{-23}\,J/K)\times T}$$

Boltzman's constant

temperature in *kelvin*

Ambient temperature $=290K$

In *dBm*: $$\boxed{N_{0dB} = -174\,dBm/Hz}$$

Example:

- Temperature = 290K (ambient)
- Bandwidth = 2.0MHz

Noise Power =

$$\left(N_0 B\right)_{dB} = -174 dBm / Hz + 10 \log_{10} 2 \times 10^6 = -111 dBm$$

2.3 Signal to Noise Ratio and "Eb/N0"

$$\boxed{SNR = \left(\frac{F_S}{B}\right)\left(\frac{E_S}{N_0}\right) = N_b\left(\frac{F_S}{B}\right)\left(\frac{E_b}{N_0}\right)}$$

F_S = symbol rate (1/sec)

B = bandwidth (Hz = 1/sec) $\geq F_S$

N_0 = noise power spectral density

E_S = Energy per symbol (Joules)

E_b = Energy per bit (Joules)

N_b = bits per symbol

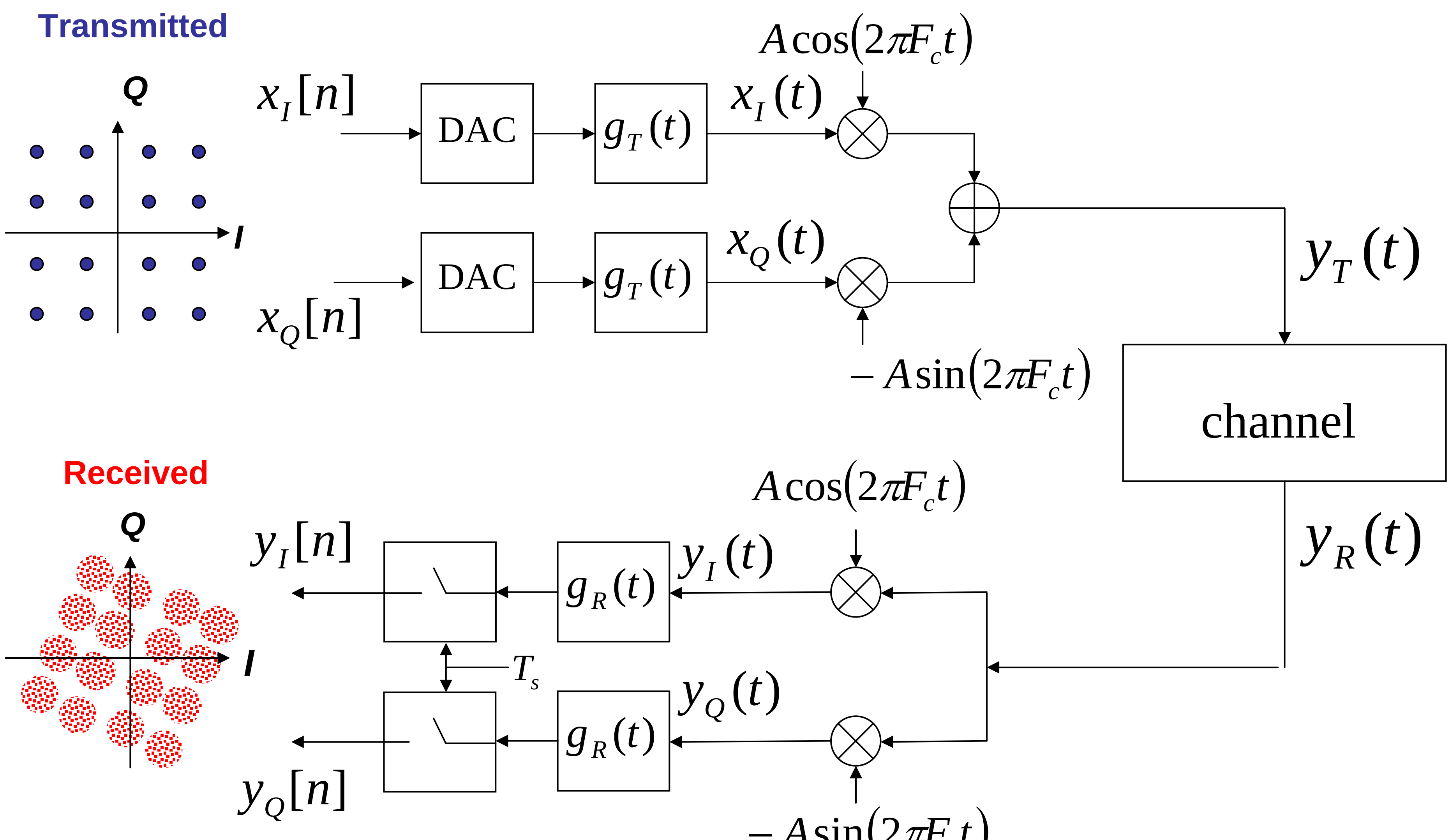
3. Complex Baseband Representation

3.1 Complex Signals

3.2 Baseband Channel Representation

3.1 Complex Signals

Symbols:

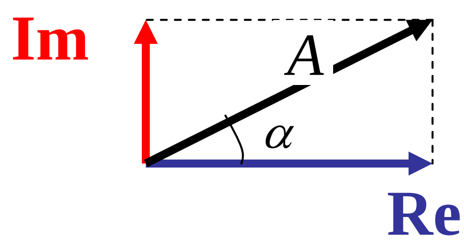


Recall: Complex Numbers and Complex Exponentials

amplitude phase (angle)

$\boxed{j = \sqrt{-1}}$

$$Ae^{j\alpha} = \underbrace{A\cos(\alpha)}_{\text{Real Part}} + j\underbrace{A\sin(\alpha)}_{\text{Imaginary Part}}$$

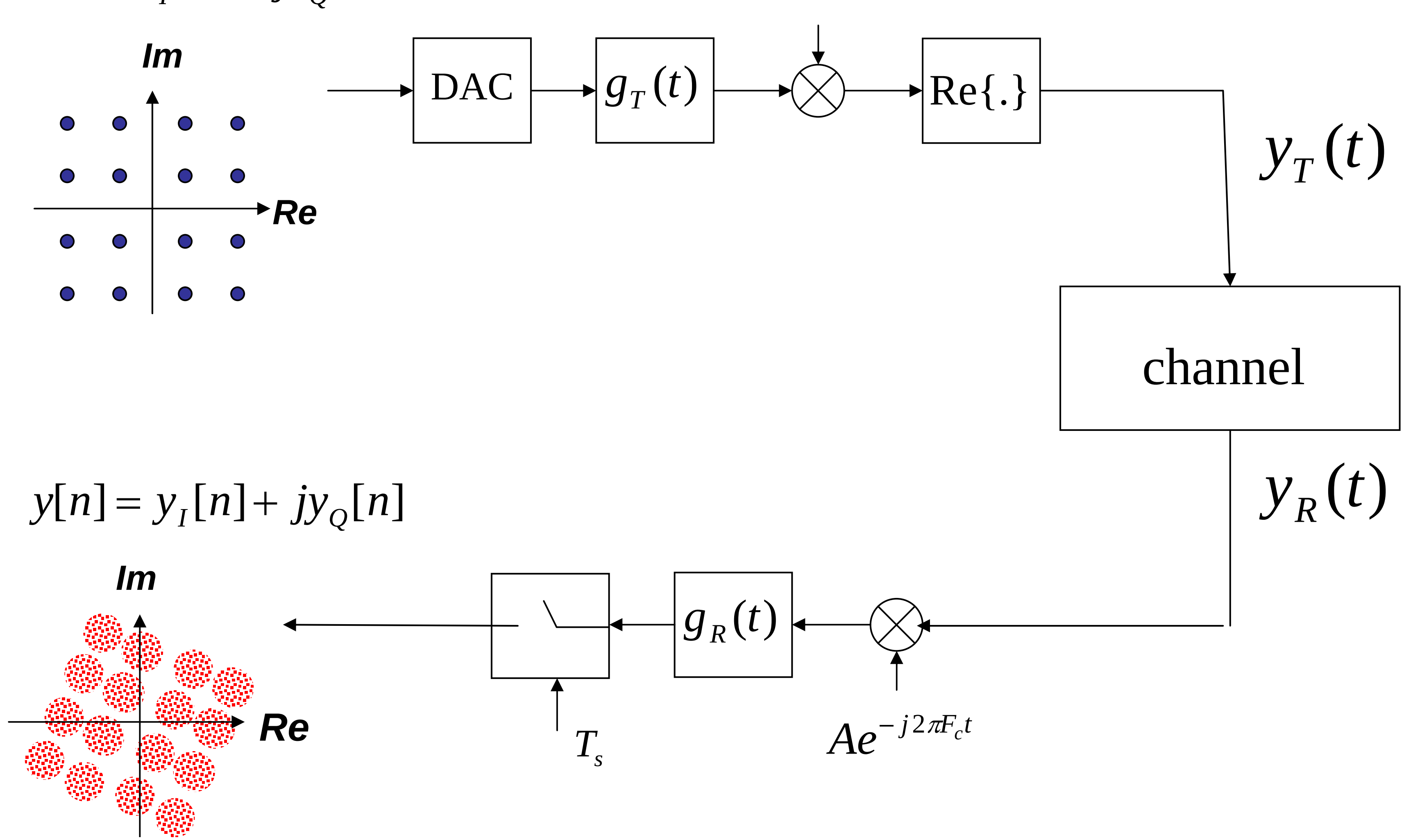


As a consequence:

$$\boxed{Ae^{j2\pi F_C t} = A\cos(2\pi F_C t) + jA\sin(2\pi F_C t)}$$

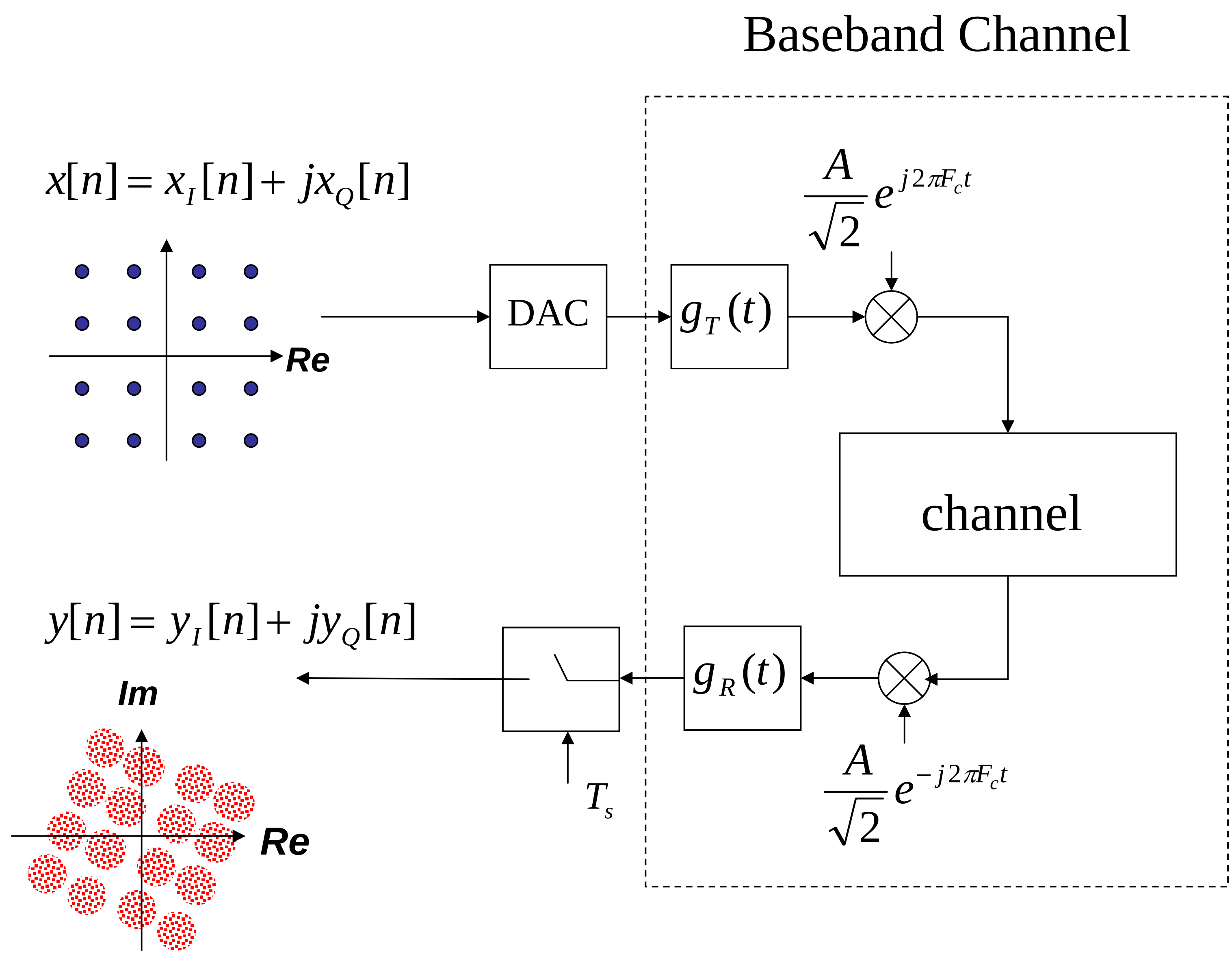
It is easier to define one complex signal which combines *In Phase* and *Quadrature* components:

$$x[n] = x_I[n] + jx_Q[n]$$

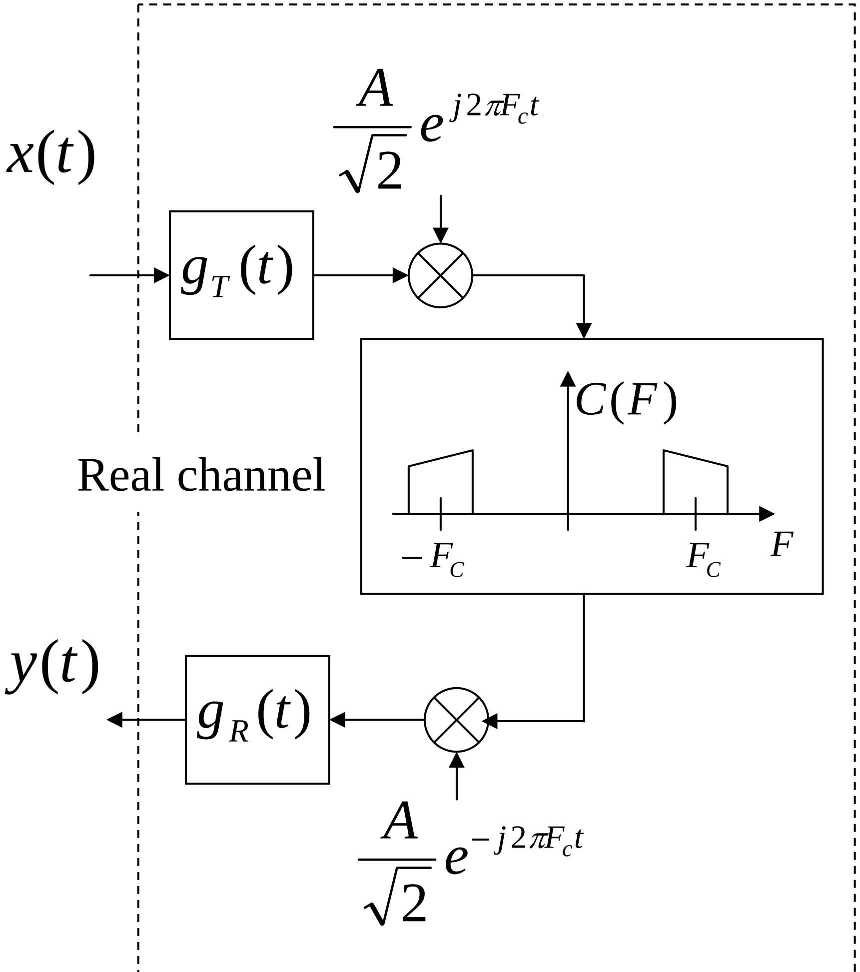


$$y[n] = y_I[n] + jy_Q[n]$$

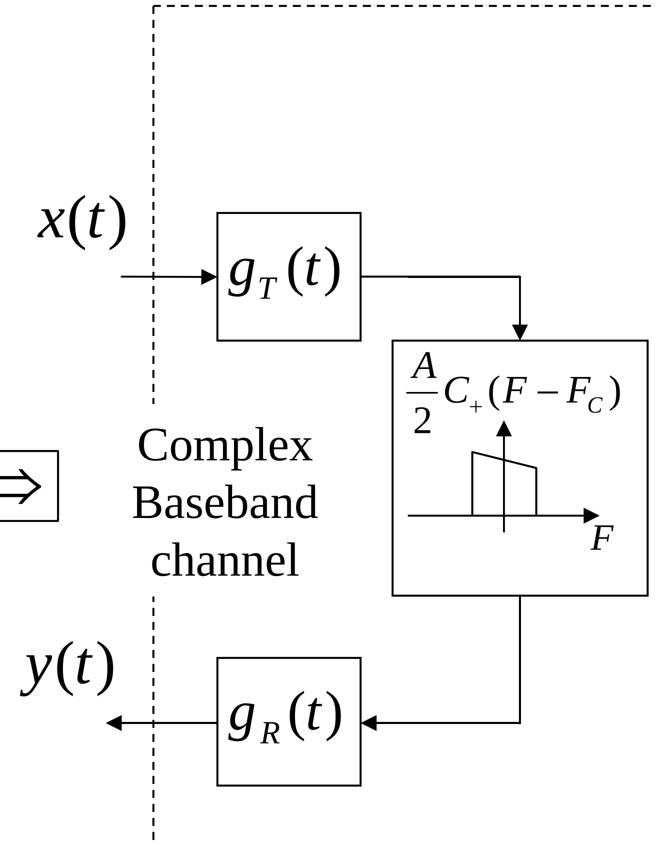
If the carrier frequency is larger than the bandwidth of the filters, then it can be simplified as



3.2 Baseband Channel Representation

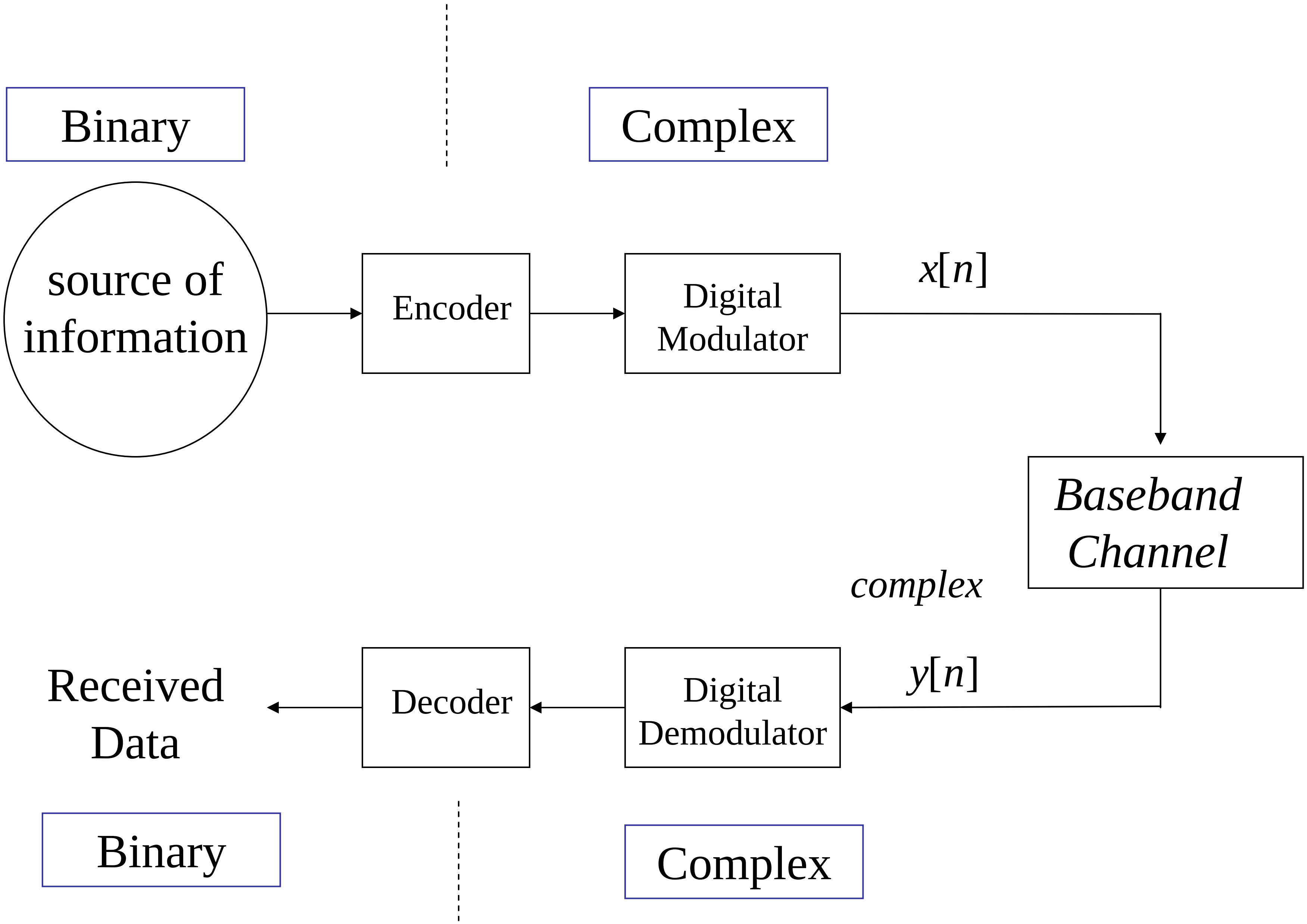


$\Leftarrow$ same $\Rightarrow$



<u>Advantage</u>: all signals at lower frequencies, therefore much easier to simulate and analyze.

Digital Transmitter/Receiver with Baseband Channel:



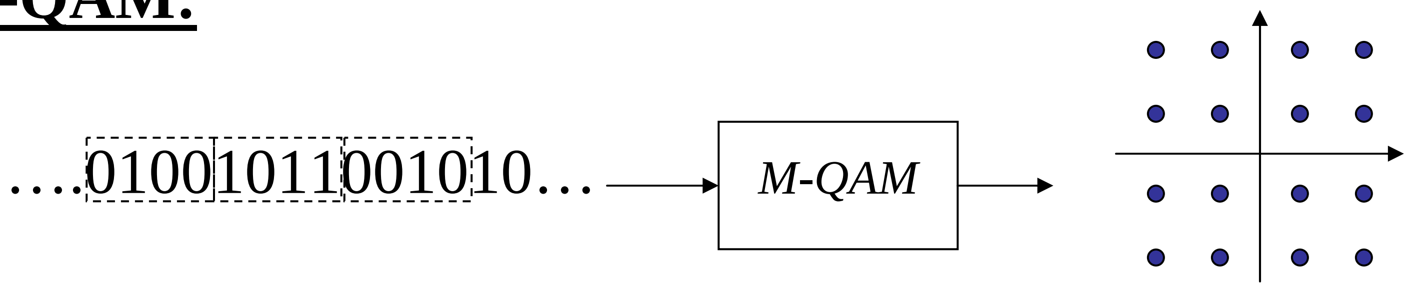
4. Bit Error Probabilities for M-QAM Modulation

4.1 "Eb/N0" and SNR for MQAM Signals

4.2 Probability of Symbol Error in Additive White Gaussian Noise (AWGN) Channel;

4.1 “Eb/N0” and SNR for MQAM Signals

M-QAM:



# of bits per symbol

Energy per bit

$$SNR = \frac{E_S / T_S}{N_0 B} = \left(\frac{\log_2(M) E_b}{N_0} \right) \left(\frac{F_S}{B} \right)$$

Therefore, if $F_S \cong B$:

$$\boxed{\frac{E_b}{N_0} = \left(\frac{1}{\log_2(M)}\right) SNR}$$

Significance: the error probabilities for MQAM are functions of E_b / N_0

<u>Example.</u>

<u>Given:</u>

Transmitted Power:	100mW
Bandwidth:	1.75MHz
Channel Attenuation:	-120dB
Modulation:	QPSK (2 bits/symbol)

<u>Assume</u>: Thermal Noise only at ambient temperature

<u>Compute</u>: E_b / N_0

Solution:

1. Energy per Symbol at the transmitter

$$E_S = 100mW / 1.75MHz = 57.14 \times 10^{-6} mW / Hz = -42.4 dBm / Hz$$

2. Energy per Symbols at the receiver

$$E_S = -42.4 - 120 = -162.4 dBm / Hz$$

3. SNR at the receiver

$$SNR = \left(\frac{E_S}{N_0}\right)_{dB} = -162.4 - (-174) = 11.6 dB$$

4. Finally $\boxed{\left(E_b / N_0\right)_{dB} = 11.6 - 10\log_{10} 2 = 8.6 dB}$

4.2 Probability of Symbol Error in AWGN

BPSK: $$P_S = Q\left(\sqrt{2\tfrac{E_b}{N_0}}\right)$$

4-QAM: $$P_S \approx 2Q\left(\sqrt{2\tfrac{E_b}{N_0}}\right)$$

M-QAM: $$P_S \approx 4Q\left(\sqrt{\left(\frac{3\log_2 M}{M-1}\right)\left(\frac{E_b}{N_0}\right)}\right)$$

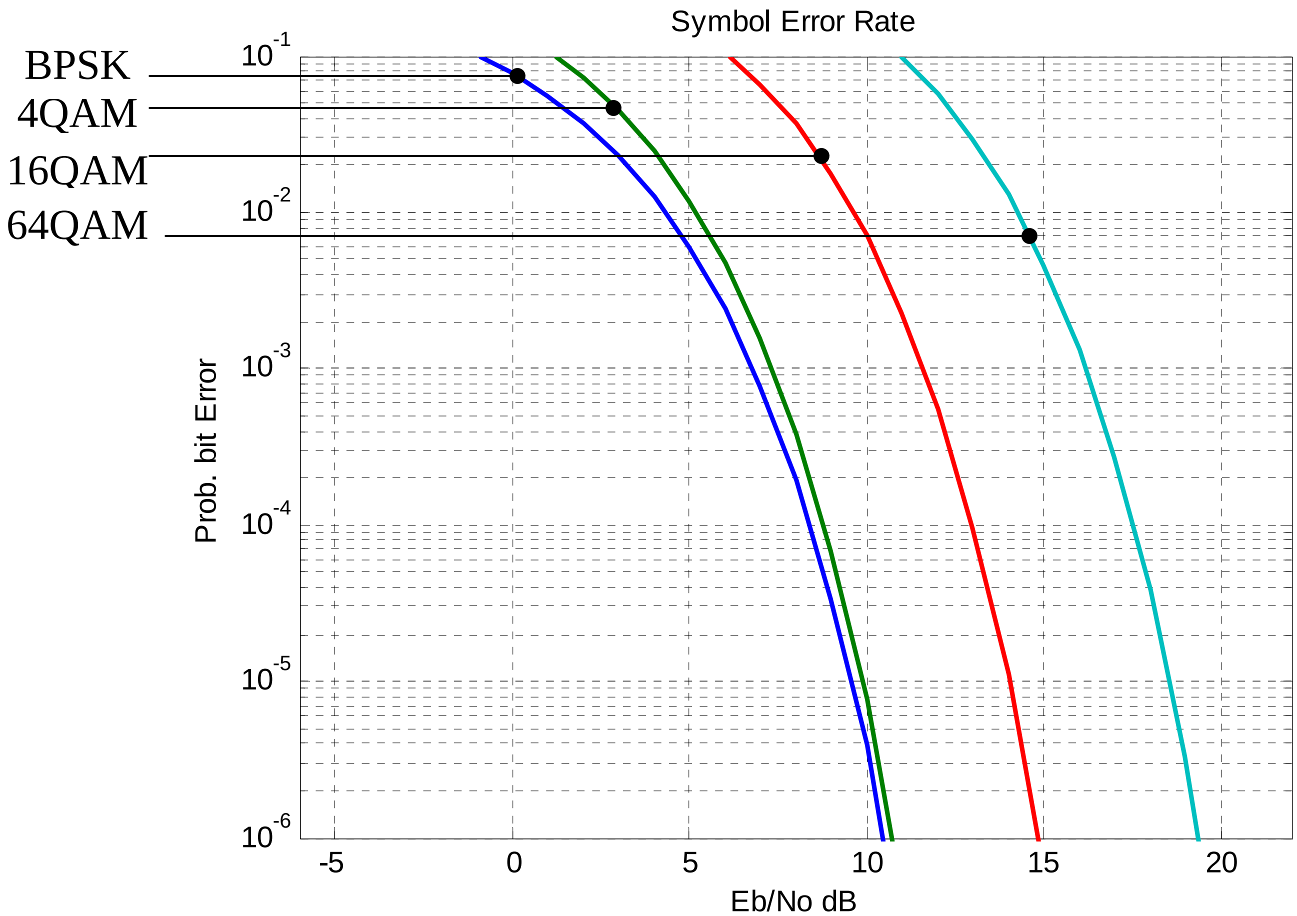
Probability of Bit Error with Gray Coding:

M-QAM: $$P_b \approx \frac{1}{\log_2(M)} P_S$$

where $$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-t^2 dt}$$

$$= \tfrac{1}{2} erfc\left(\frac{x}{\sqrt{2}}\right)$$

Symbol Error Rates (Exact Values)



Example.

Same as previous example

Compute: a. Symbol Error Rate, b. Bit Error Rate

Solution:

1. From the previous Example $E_b / N_0 = 8.6dB$
2. Symbol Error Rate: 10^{-4} errors per symbol (see graphs)
3. Bit Error Rate: $10^{-4} / 2 = 5 \times 10^{-5}$ errors per bit

5. Simulink Implementation

5.1 Fundamentals of Simulink

5.2 Digital Communications in AWGN: Simulink Implementation;

5.3 Example

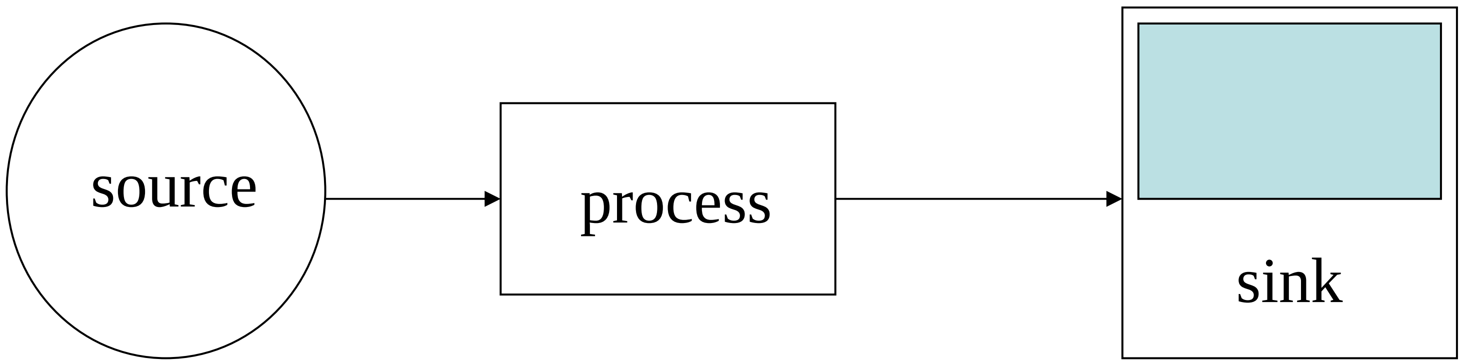
5.1 Fundamentals of Simulink

Simulink has three classes of blocksets:

• sources (outputs only)

• processes (inputs/outputs)

• sinks (inputs only)



• generate data

• read data

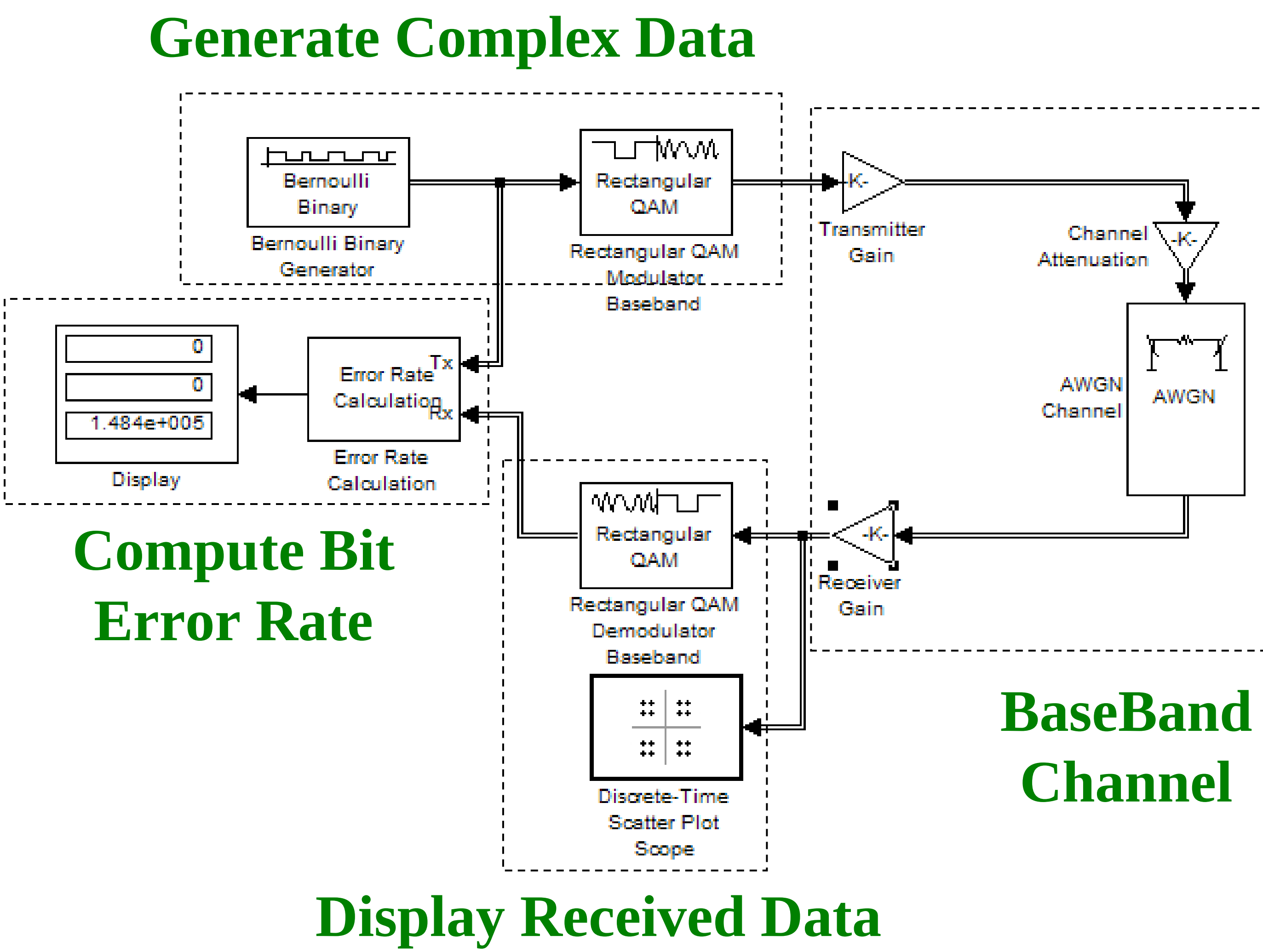
• import files …

• display results

• plots, scopes…

• send data to devices

5.2 Digital Communications in AWGN: Simulink Implementation



Simulink Model: **AWGN_no_coding.mdl**

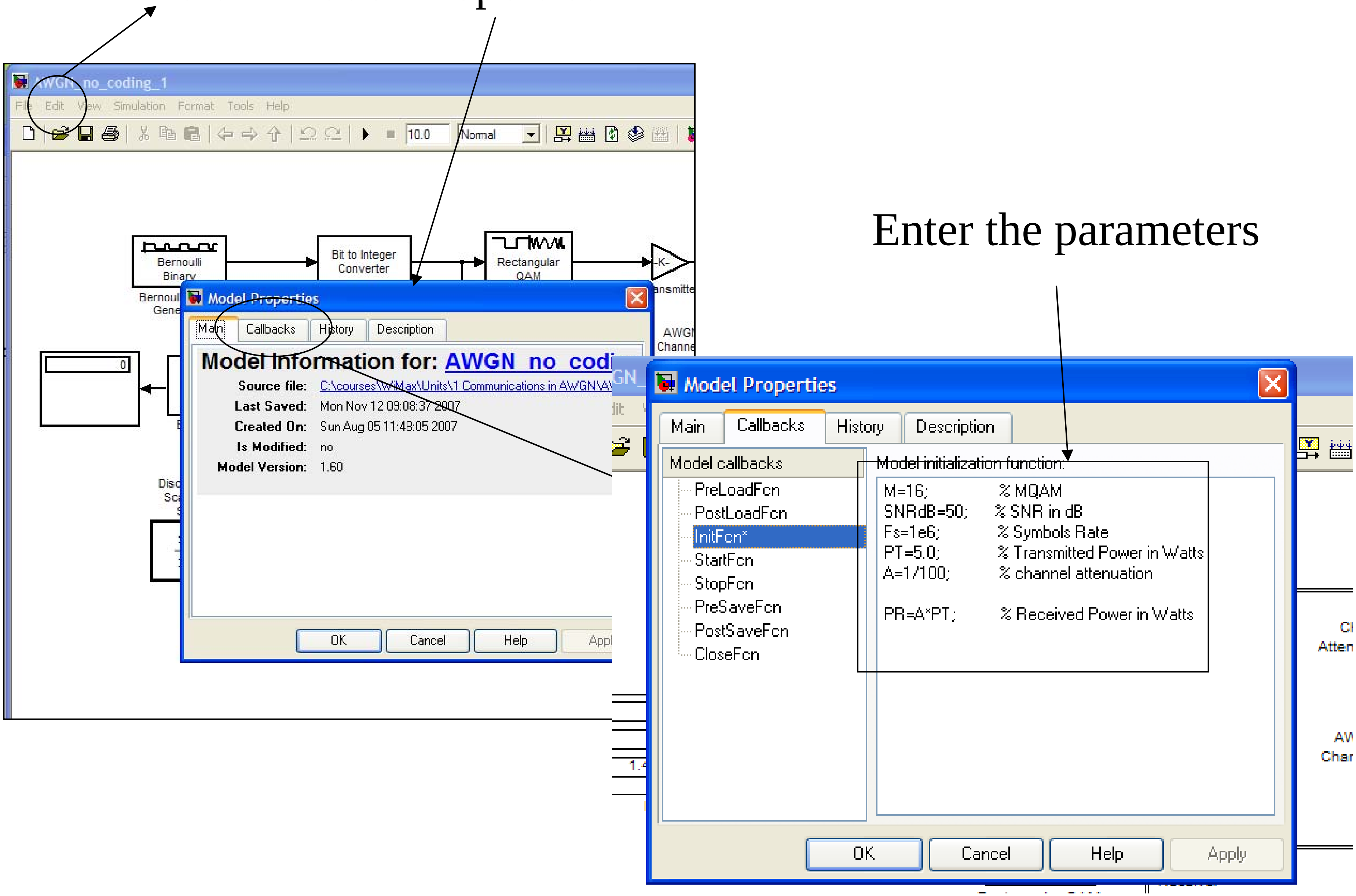
5.3 Example

Let's simulate a Digital Communications System with the following parameters:

```
M=4;             % MQAM modulation
Fs=10^6;         % symbol rate (1/sec)
SNRdB=20;        % SNR in dB's
PT=5;            % Transmitted Power in Watts
A=1/100;         % Channel Attenuation
PR=A*PT;         % Received Power
```

Edit > Model Properties

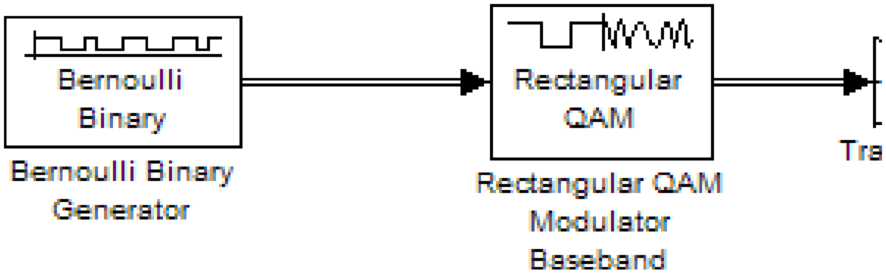
Enter the parameters



Generate Complex Data

Parameters:

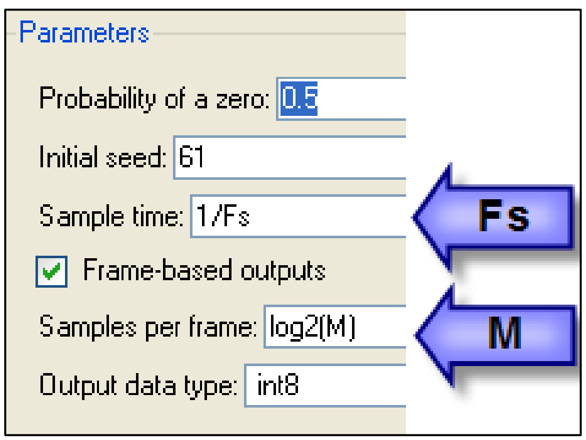
- **M**-QAM
- **Fs** symbols/sec



Communications Sources

>Random Data Sources

>>**Bernoulli Binary**

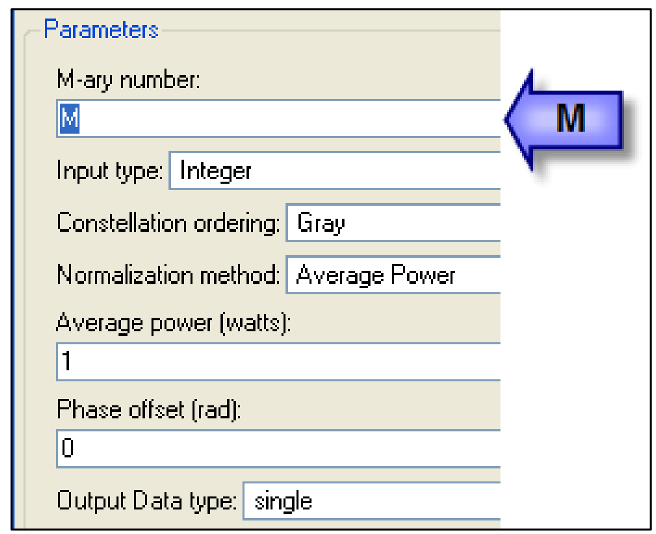


Modulation

>Digital Baseband Mod

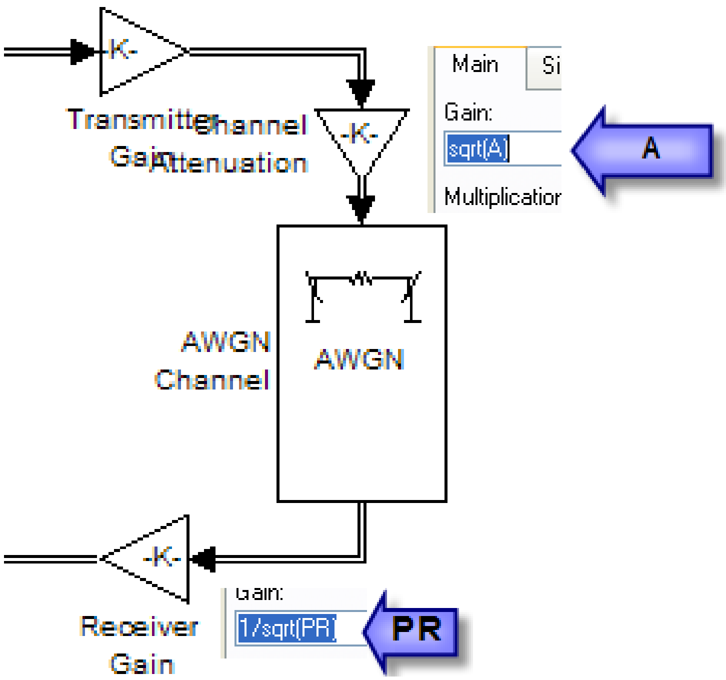
>>AM

>>>**Rectangular QAM**

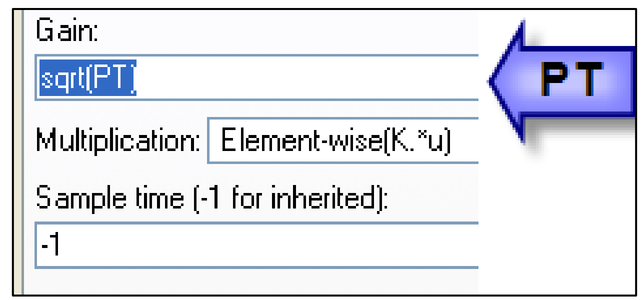


Baseband Channel

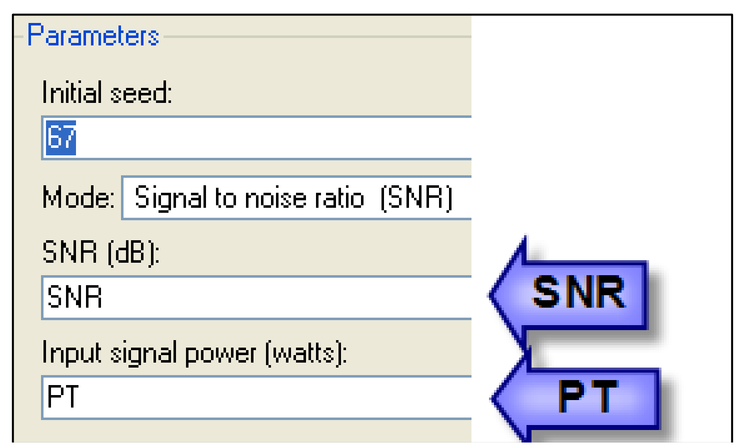
$$y[n] = \frac{1}{\sqrt{P_R}} A \times \left(\sqrt{P_T}\, x[n] + \sqrt{\frac{P_T}{SNR}}\, w[n] \right)$$



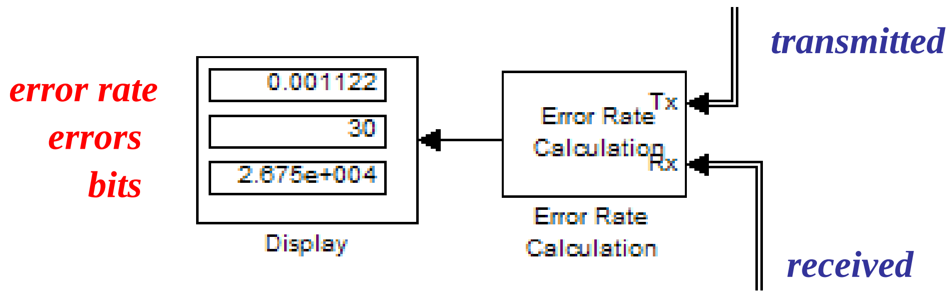
Transmitter Gain




```
Channels
> AWGN Channel
```



Analysis of Data by Computer Simulation



Blocks:

- `Comms Sinks > Error Rates Calculation`
- `Simulink > Sinks > Display`

Lab2: Digital Communications Fundamentals

Given a digital Communication System defined by the following parameters:

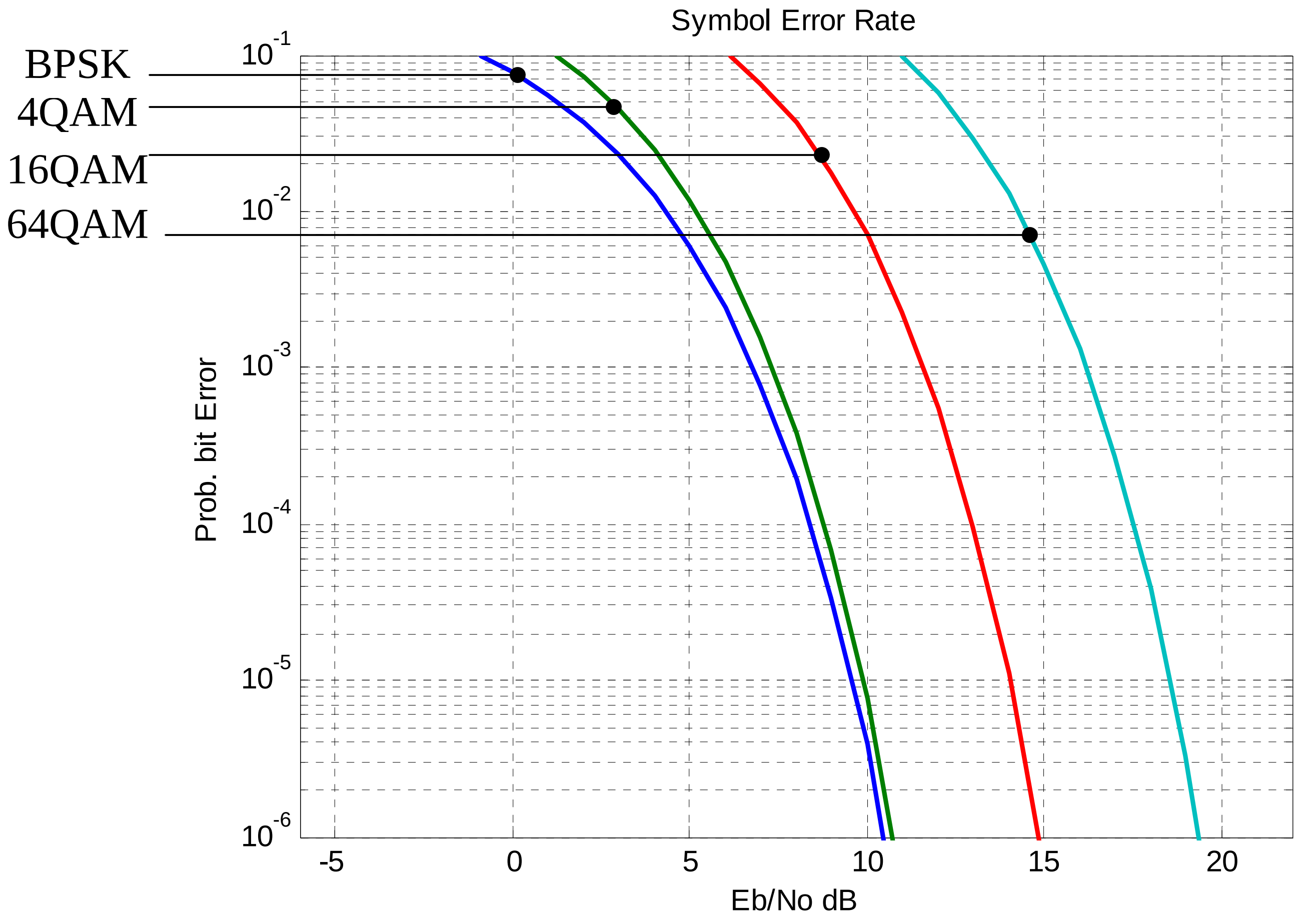
```
M=4;           % MQAM modulation
Fs=10^6;       % symbol rate (1/sec)
PT=2;          % Transmitted Power in Watts
A=1/50;        % Channel Attenuation
PR=A*PT;       % Received Power
```

1. Simulate the system for the following values of the Signal to Noise Ratio: SNR=5,10,20dB
2. For each case determine the probability of bit error experimentally
3. Compare with the theoretical values

References: **AWGN_no_coding.mdl**

bit_error.m

Probability of Symbol Error



Probability of Bit Error= Probability of Symbol Error / log2(M)

3. Channel Models

1. **Introduction and Channel Losses**
2. **Models of Fading Channels**
3. **Channel Parameterization**
4. **Estimation of Channel Parameters from Data**

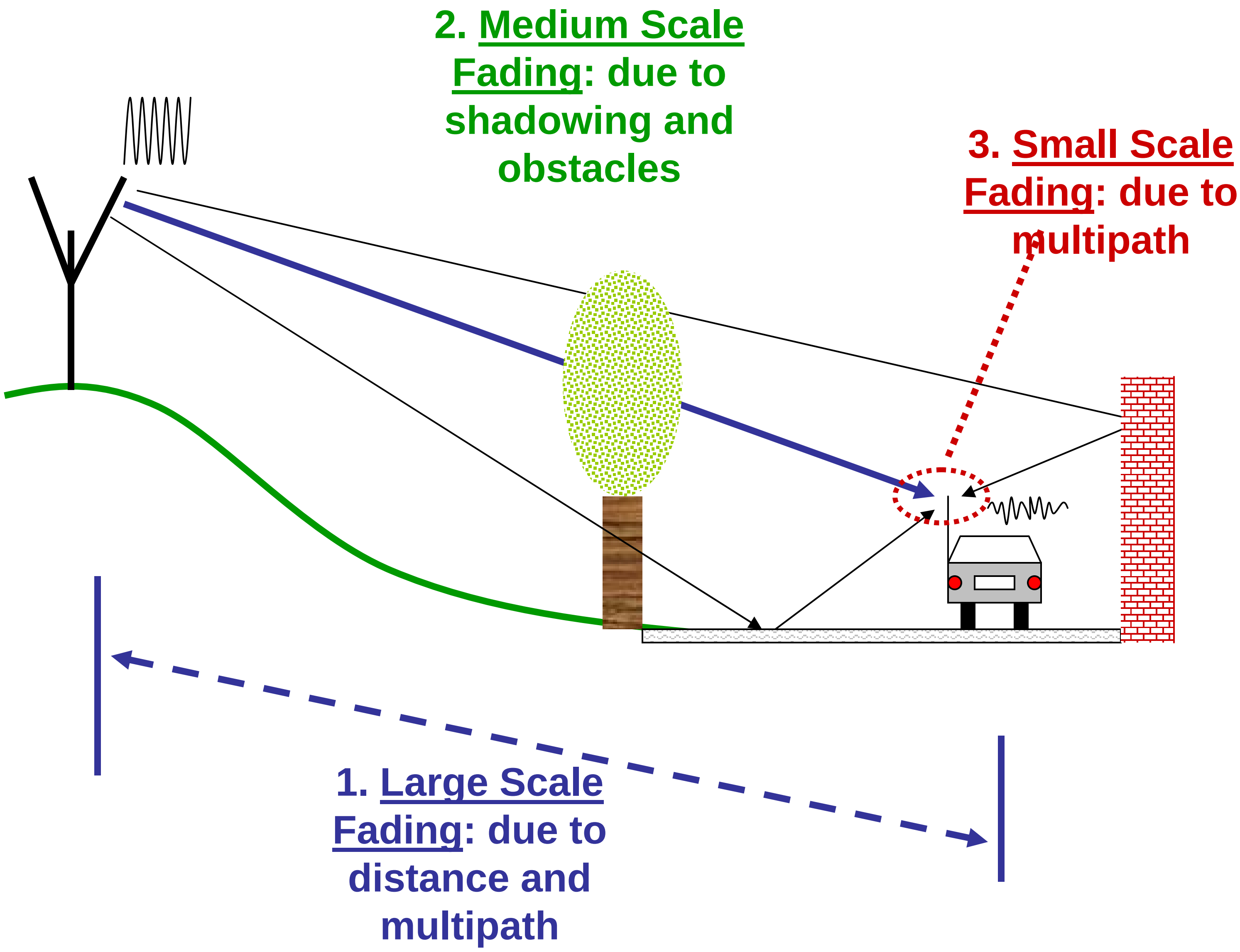
1. Introduction and Channel Losses

1.1. Large Scale fading: Free Space Losses
1.2. Medium Scale Fading: Shadowing
1.3. Small Scale Fading: Multipath

References:

A. Goldsmith, Wireless Communications, Cambridge Univ. Press, 2005 – Chapter 2.

Signal Losses due to three Effects:



Wireless Channel

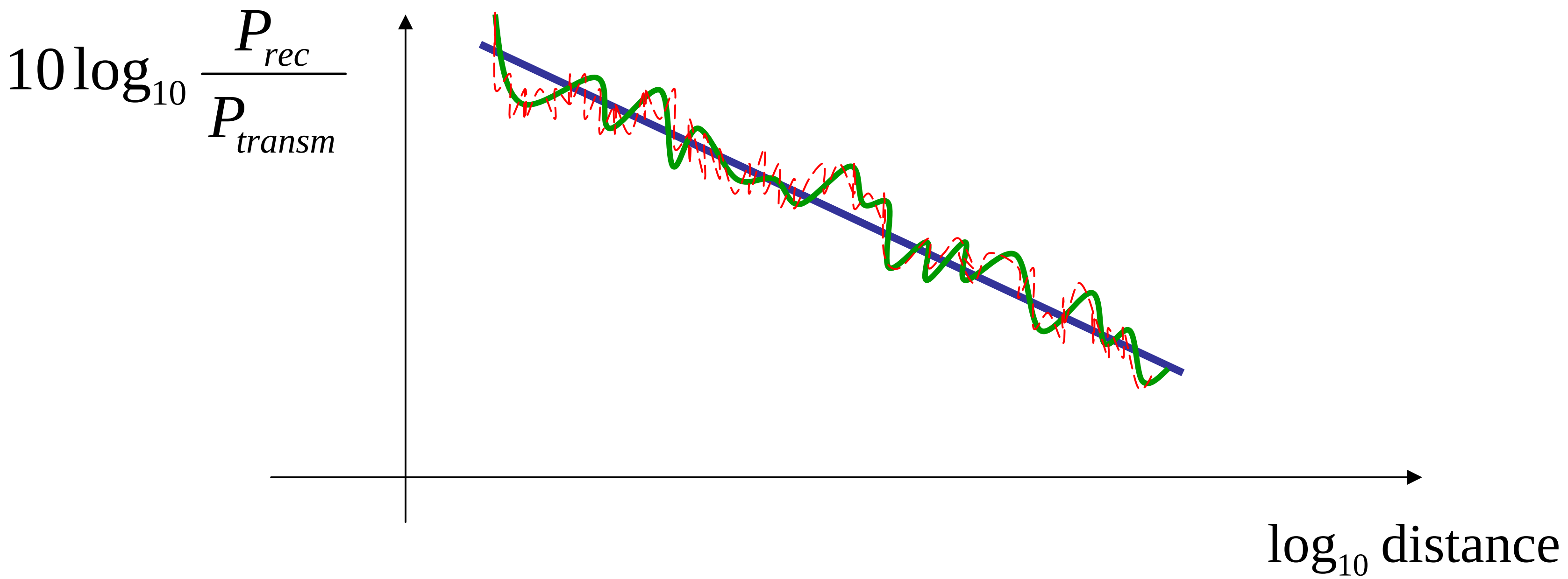
Frequencies of Interest: in the UHF (.3GHz – 3GHz) and SHF (3GHz – 30 GHz) bands;

Several Effects:

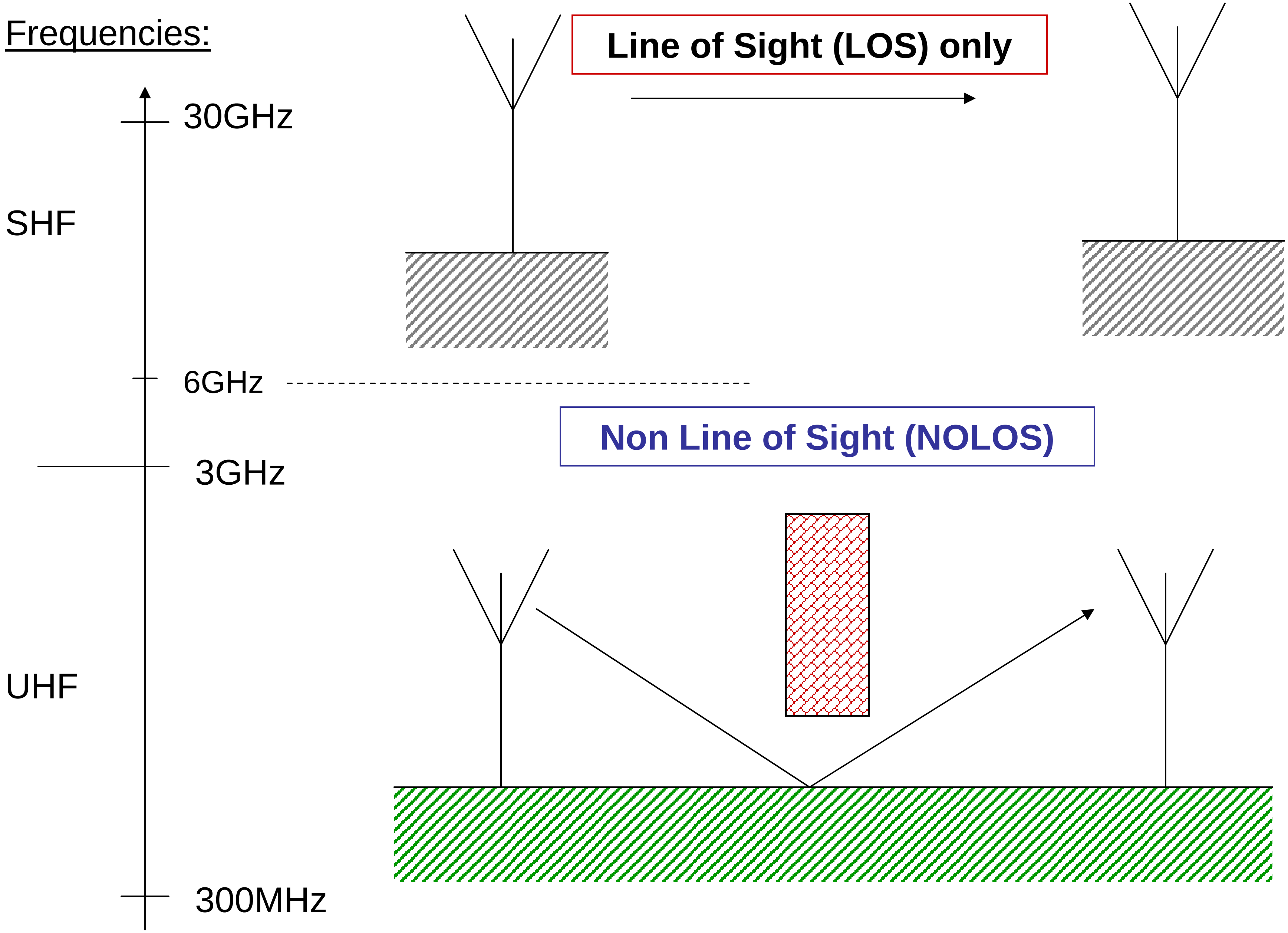
• **Path Loss due to dissipation of energy: it depends on distance only**

• **Shadowing due to obstacles such as buildings, trees, walls. Is caused by absorption, reflection, scattering …**

• **Self-Interference due to Multipath.**

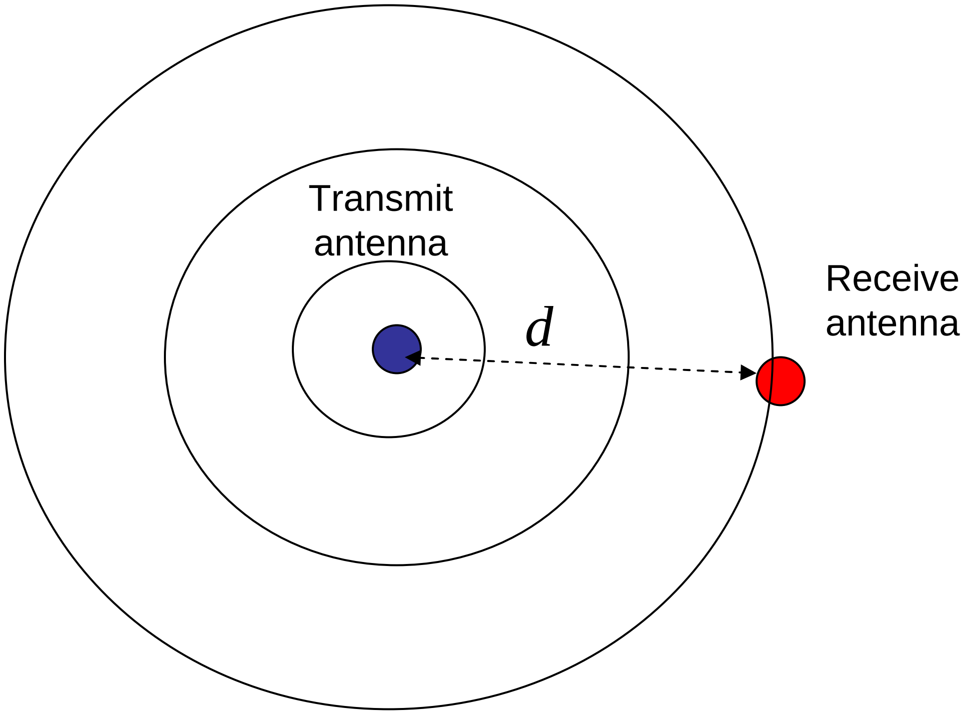


Line of Sight and Frequencies



1.1. Large Scale Propagation: Free Space

Path Loss due to Free Space Propagation:



For *isotropic* antennas:

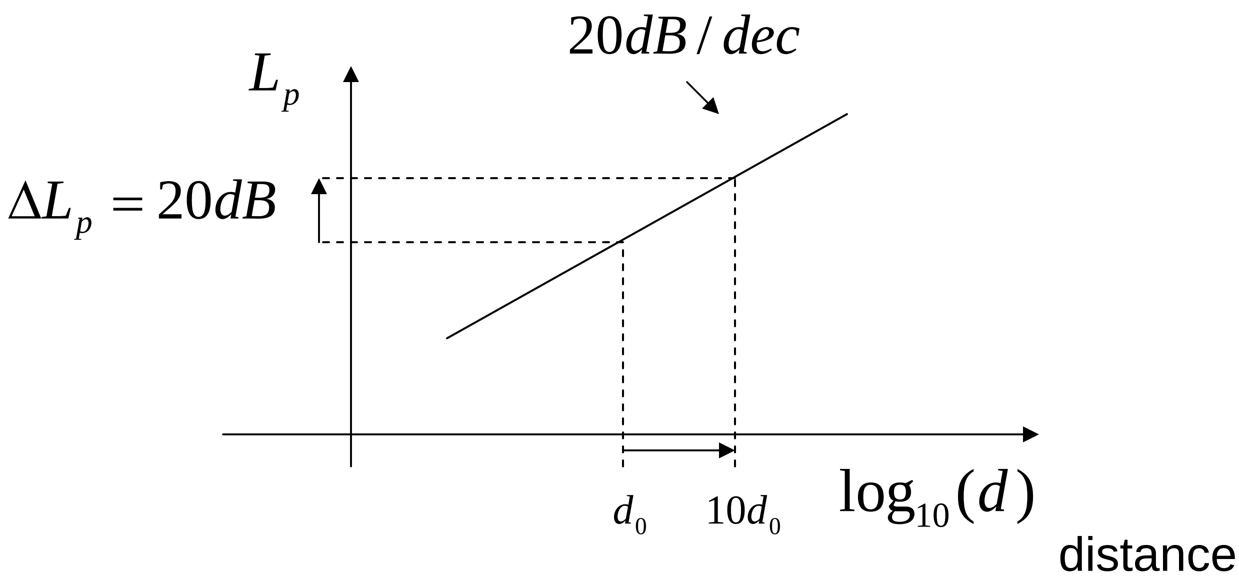
$$\boxed{P_{rec} = \left(\frac{\lambda}{4\pi d} \right)^2 P_{transm}}$$

wavelength $\lambda = \dfrac{c}{F}$

Path Loss in dB:

$$\boxed{L = 10\log_{10}\left(\frac{P_{transm}}{P_{rec}} \right) = 20\log_{10}(F(MHz)) + 20\log_{10}(d(km)) + 32.45}$$

Free Space Attenuation



$$L_{p1} - L_{p2} = 20dB \quad \Rightarrow \quad P_{rec1} = P_{rec2}/100$$

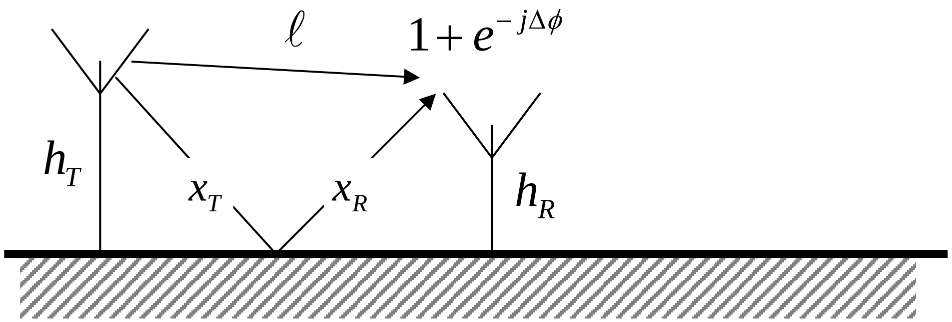
<u>Valid for:</u>

- Satellite Communicationss
- Point to Point LOS Microwave
- Reference for Path Loss Models

Multipath Models: Two Ray Reflection

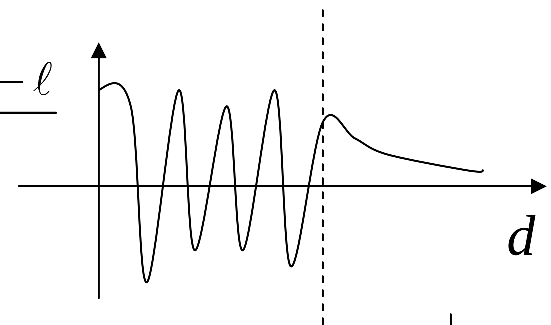
Typical of open environments such as rural roads

• Small to medium distances

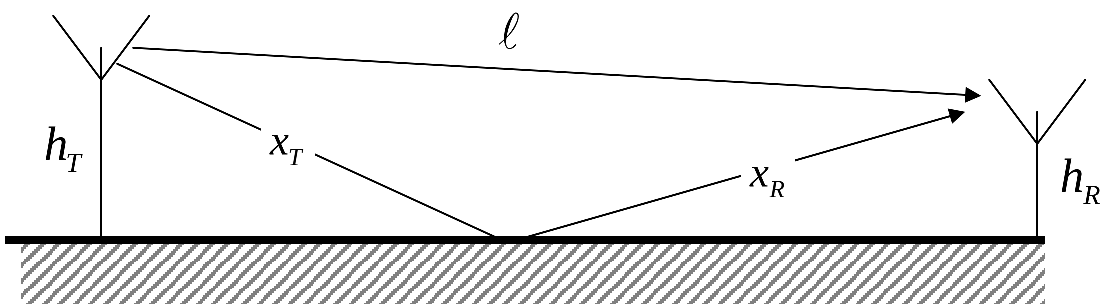


<u>constructive/distructive interference</u>

$$\Delta\phi = 2\pi \frac{x_T + x_R - \ell}{\lambda}$$

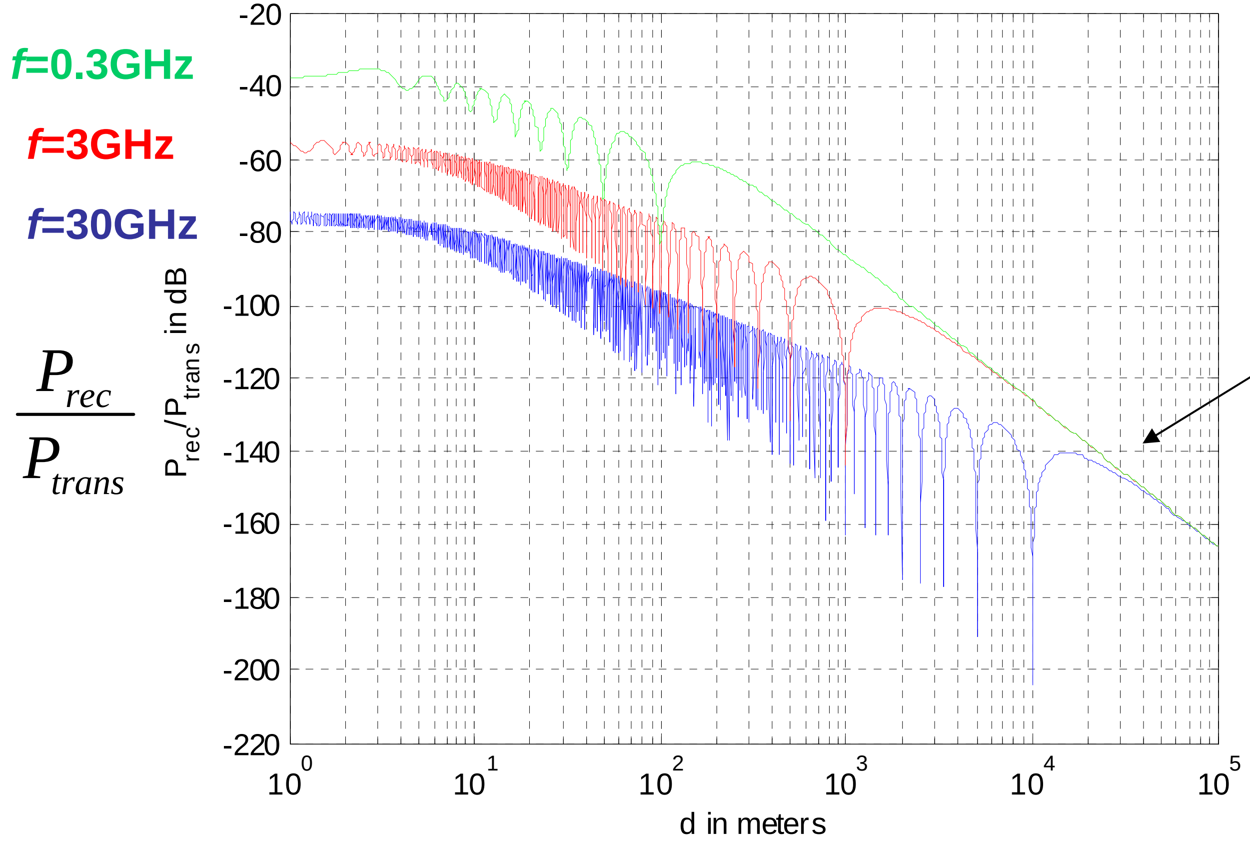


• Large distances



<u>monotonic</u>

Two Ray Model Power Received



$$\frac{P_{rec}}{P_{trans}}$$

$$-40dB/dec$$

independent of frequency

$$\frac{P_{rec}}{P_{trans}} \approx \frac{h_T^2 h_R^2}{d^4}$$

Assume: reflecting surface a pure dielectric

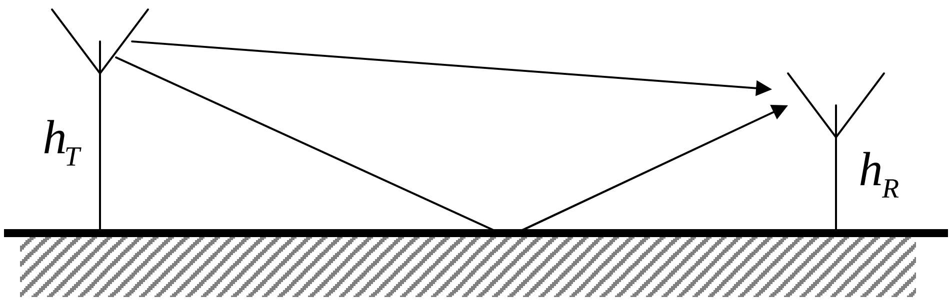
Compare the two:

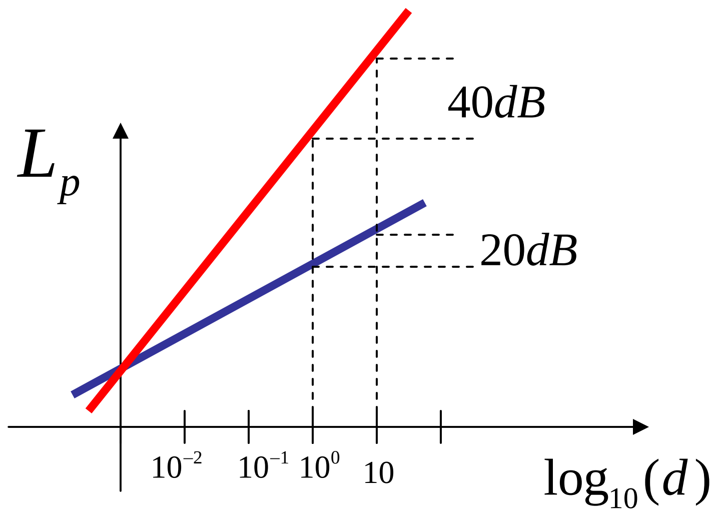
Free Space:

$$\boxed{L_p = -20\log_{10}(\pi\lambda/4) + 20\log_{10}(d)}$$

Two Ray Approximation:

$$\boxed{L_p = -20\log_{10}(h_T h_R) + 40\log_{10}(d)}$$





2. Medium Scale Fading: Losses due to Buildings, Trees, Hills, Walls …

The Power Loss in dB is random:

$$L_p = E\{L_p\} + \chi$$

expected value

random, zero mean

approximately gaussian with

$$\sigma \approx 6 - 12\ dB$$

Average Loss

Free space loss at reference distance

$$E\{L_p\} = 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + L_0 \quad \text{dB}$$

Path loss exponent

Reference distance

- indoor 1-10m
- outdoor 10-100m

Values for Exponent γ :

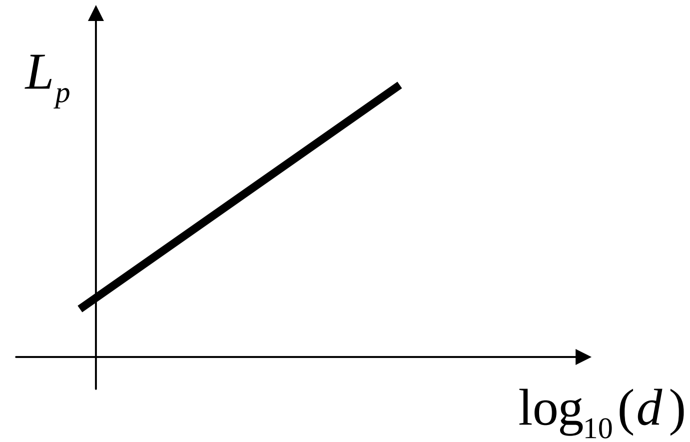
Free Space	2
Urban	2.7-3.5
Indoors (LOS)	1.6-1.8
Indoors(NLOS)	4-6

Empirical Models for Propagation Losses to Environment

- **Okumura**: urban macrocells 1-100km, frequencies 0.15-1.5GHz, BS antenna 30-100m high;
- **Hata**: similar to Okumura, but simplified
- **COST 231**: Hata model extended by European study to 2GHz

Typical: Hata Models (1980)

Frequencies 0.15-1.5GHz



Propagation Loss $L_p = \alpha + \beta \log_{10}(d)$

$$\beta = 44.9 - 6.55 \log_{10}(h)$$

urban $\alpha = \alpha_0 = 69.55 + 26.16 \log_{10}(f) - 13.82 \log_{10}(h_T) - a(h_R)$

suburban $\alpha = \alpha_0 - 2 \log_{10}^2(f/28) - 5.4$

rural $\alpha = \alpha_0 - 4.78 \log_{10}^2(f) + 18.33 \log_{10}(f) - 40.94$

where

f = frequency in MHz

h_T, h_R = transmitter antenna elevation over average terrain (in meters)

$a(h_R)$ corrective factor for receiver antenna

Receiver antenna correcting factor

Small to medium city

$$a(h_R) = \left(1.1\log_{10}(f) - 0.7\right)h_R - \left(1.56\log_{10}(f) - 0.8\right) \quad dB$$

Large city

$$a(h_R) = 3.2\left(\log_{10}(11.75h_R)\right)^2 - 4.97 \quad dB$$

COST 231 Model: Urban Model

Propagation Loss

$$L_p = \alpha + \beta \log_{10}(d) + C_M$$

$$\beta = 44.9 - 6.55 \log_{10}(h)$$

$$\alpha = \alpha_0 = 46.3 + 33.9 \log_{10}(f) - 13.82 \log_{10}(h_T) - a(h_R)$$

where

$$C_M = \begin{cases} 0\,dB & \text{medium sized city and suburbs} \\ 3\,dB & \text{metropolitan area} \end{cases}$$

Restrictions:

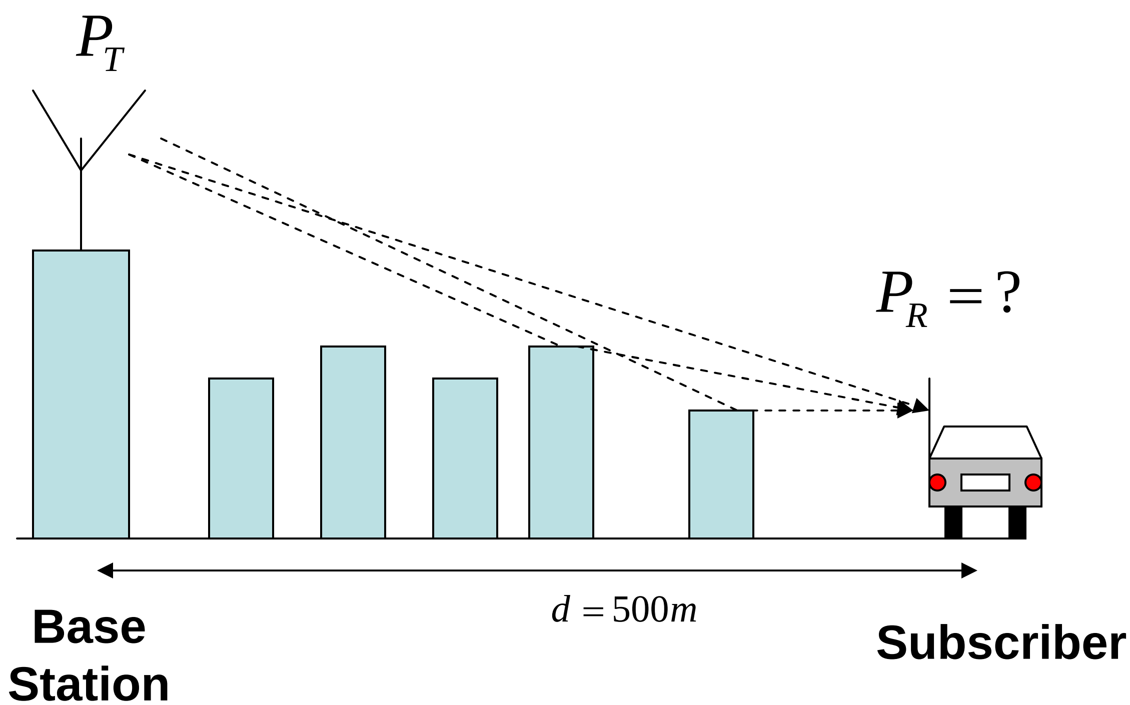
$$1.5GHz < f < 2GHz$$

$$30m < h_T < 200m$$

$$1m < h_R < 10m$$

$$1km < d < 20km$$

Example:



Given:

Channel: $$L_p = 10\gamma \log_{10}\left(\frac{d}{d_0}\right) + L_0 + \chi$$

$d_0 = 1m$

$L_0 = 45dB$

$\gamma = 3$

$\sigma = 6dB$

Transmitted Power:

$$1.0Watt = 30dBm$$

Bandwidth: 10MHz

Noise: $N_0 = -174dBm/Hz$ **thermal**

Then:

Received Power: $P_R = 30 - L_p = 30 - (10\times 3\times \log_{10} 500 + 45) + \chi = -96dBm + \chi$

Noise at the Receiver: $P_{Noise} = -174 + 10\log_{10} 10^7 = -104dBm$

SNR at the receiver: $SNR = -96 + \chi - (-104) = 8 + \chi \; dB$

random

Probability that the SNR is sufficiently large

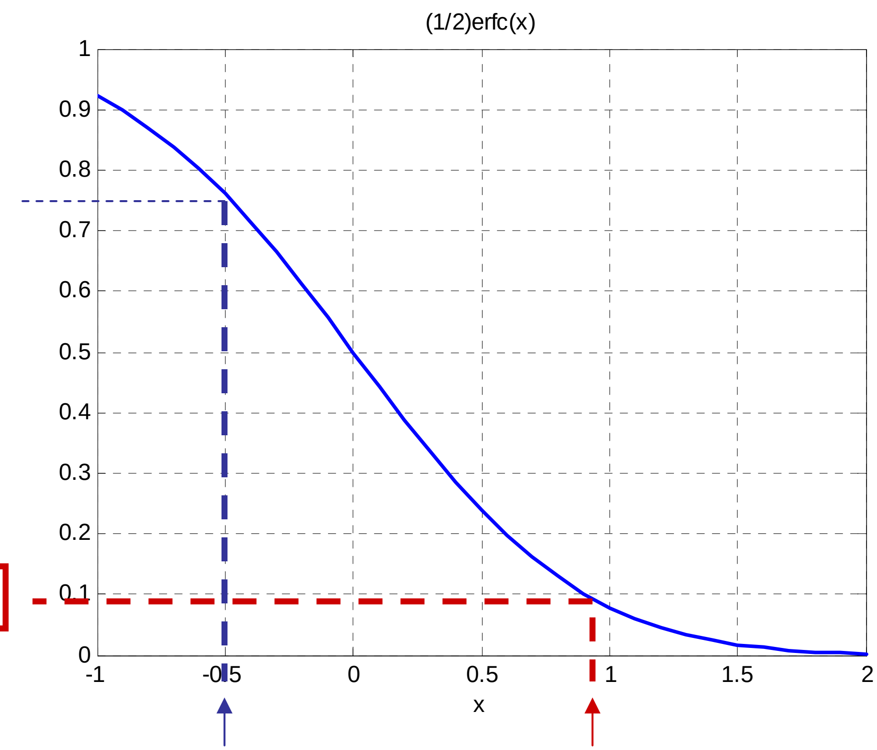
$$P(\chi > a) = \frac{1}{\sqrt{2\pi\sigma}} \int_{a}^{+\infty} e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2} dx = \frac{1}{2} erfc\left(\frac{a}{\sqrt{2\sigma}}\right)$$

Where we define $$erfc(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{+\infty} e^{-t^2} dt$$

$$P(SNR > a) = P(8 + \chi > a) = \frac{1}{2} erfc\left(\frac{a-8}{\sqrt{2}\;6}\right)$$

BPSK 75% of the time $\boxed{0.75}$

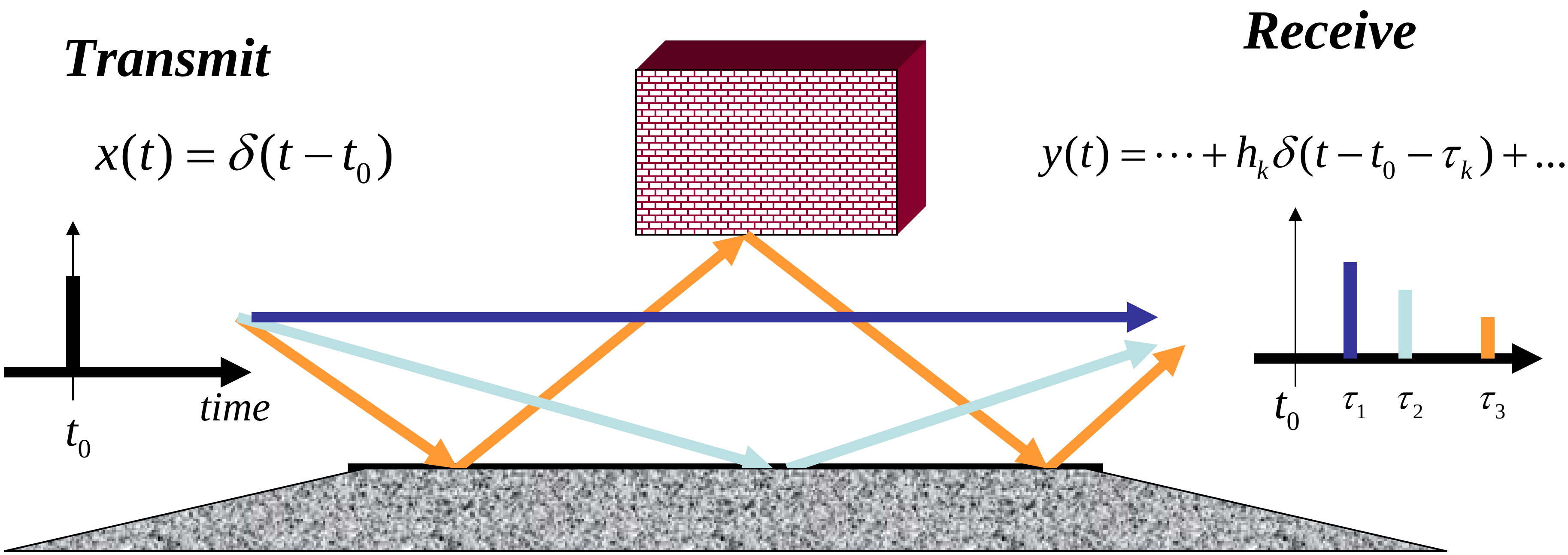
16QAM 10% of the time $\boxed{0.10}$



Required for low BER: **BPSK: a=3dB** **16QAM: a=14.7dB**

3. Small Scale Fading due to Multipath.

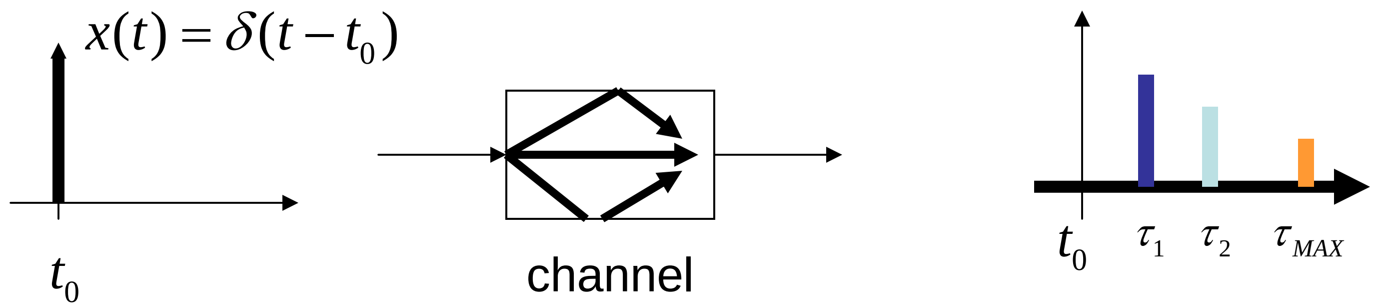
a. Spreading in Time: different paths have different lengths;



Example for 100m path difference we have a time delay

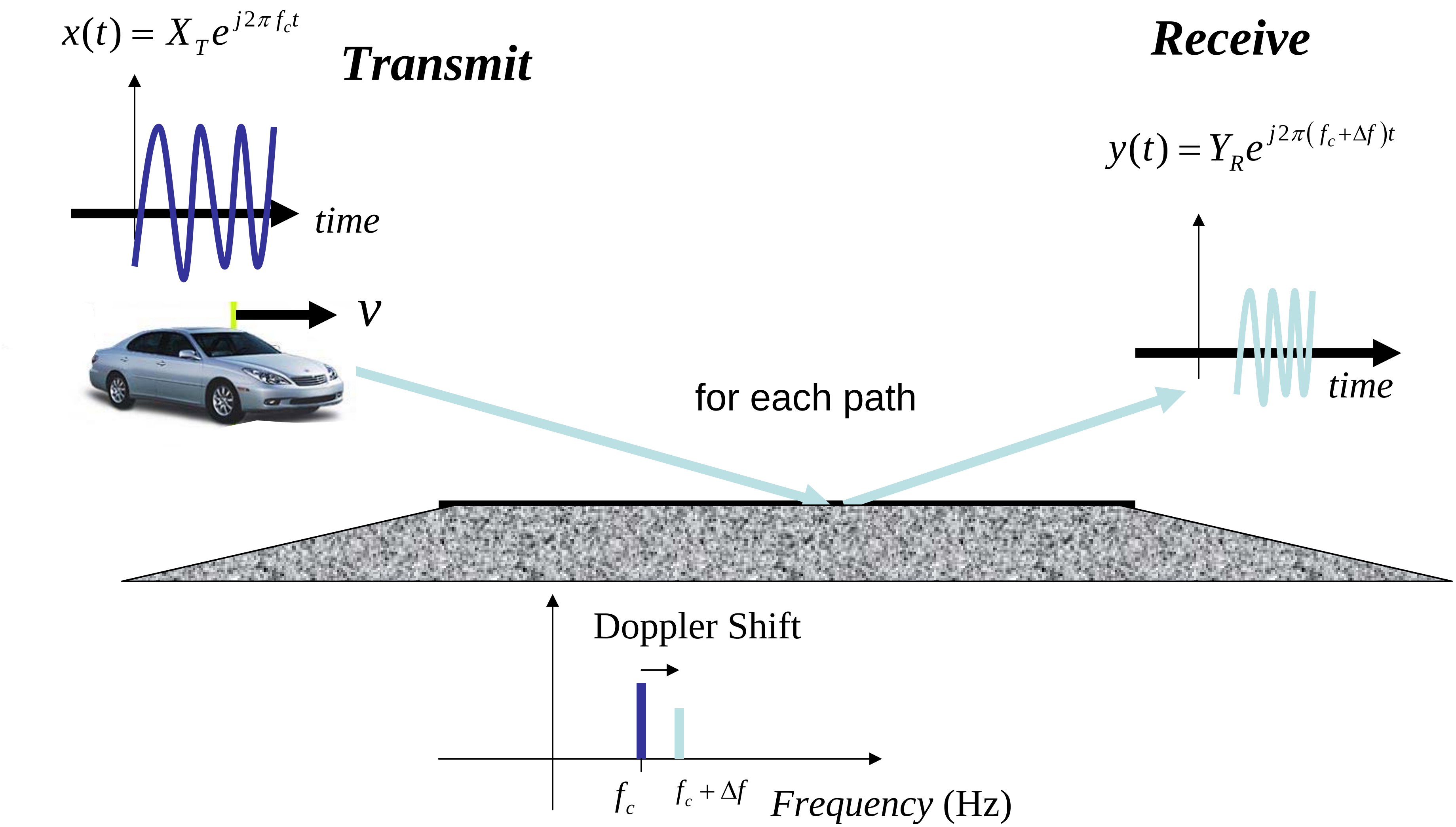
$$\tau = \frac{100}{c} = \frac{10^2}{3 \times 10^8} = \tfrac{1}{3}\,\mu\sec$$

Typical values channel time spread:

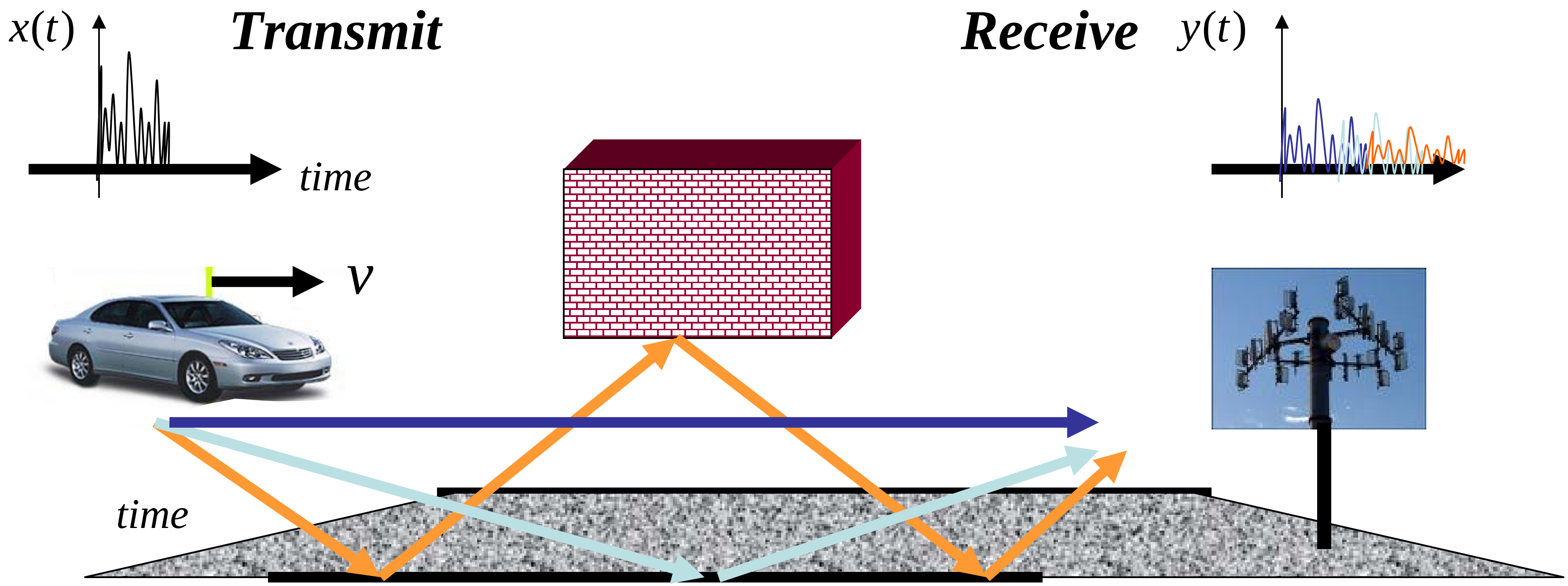


Indoor	$10 - 50\ n\sec$
Suburbs	$2 \times 10^{-1} - 2\ \mu\sec$
Urban	$1 - 3\ \mu\sec$
Hilly	3-10 μsec

b. Spreading in Frequency: motion causes frequency shift (Doppler)



Put everything together



Each path has …

… attenuation…

…shift in time …

$$y(t) = \mathrm{Re}\left\{ \underbrace{\sum_k}_{\textbf{paths}} a_k x(t-\tau_k) e^{j2\pi(F_c+\Delta F_k)(t-\tau_k)} \right\}$$

…shift in frequency …

(this causes small scale time variations)

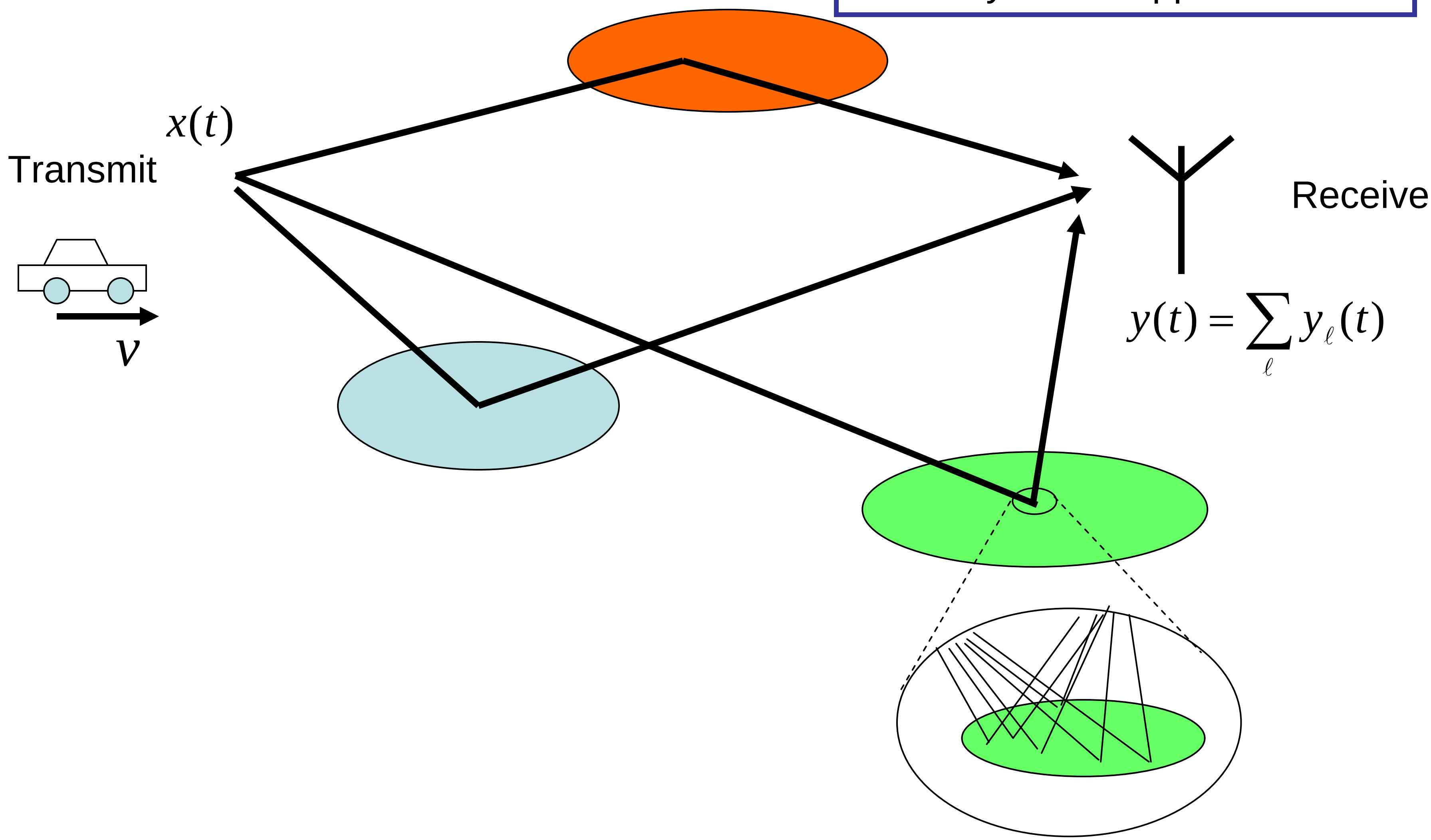
2. Models of Fading Channels

2.1. Statistical Models of Fading Channels
2.2. Non Line of Sight (Rayleigh) and Line of Sight (Rice) Channels
2.3. Simulink Example

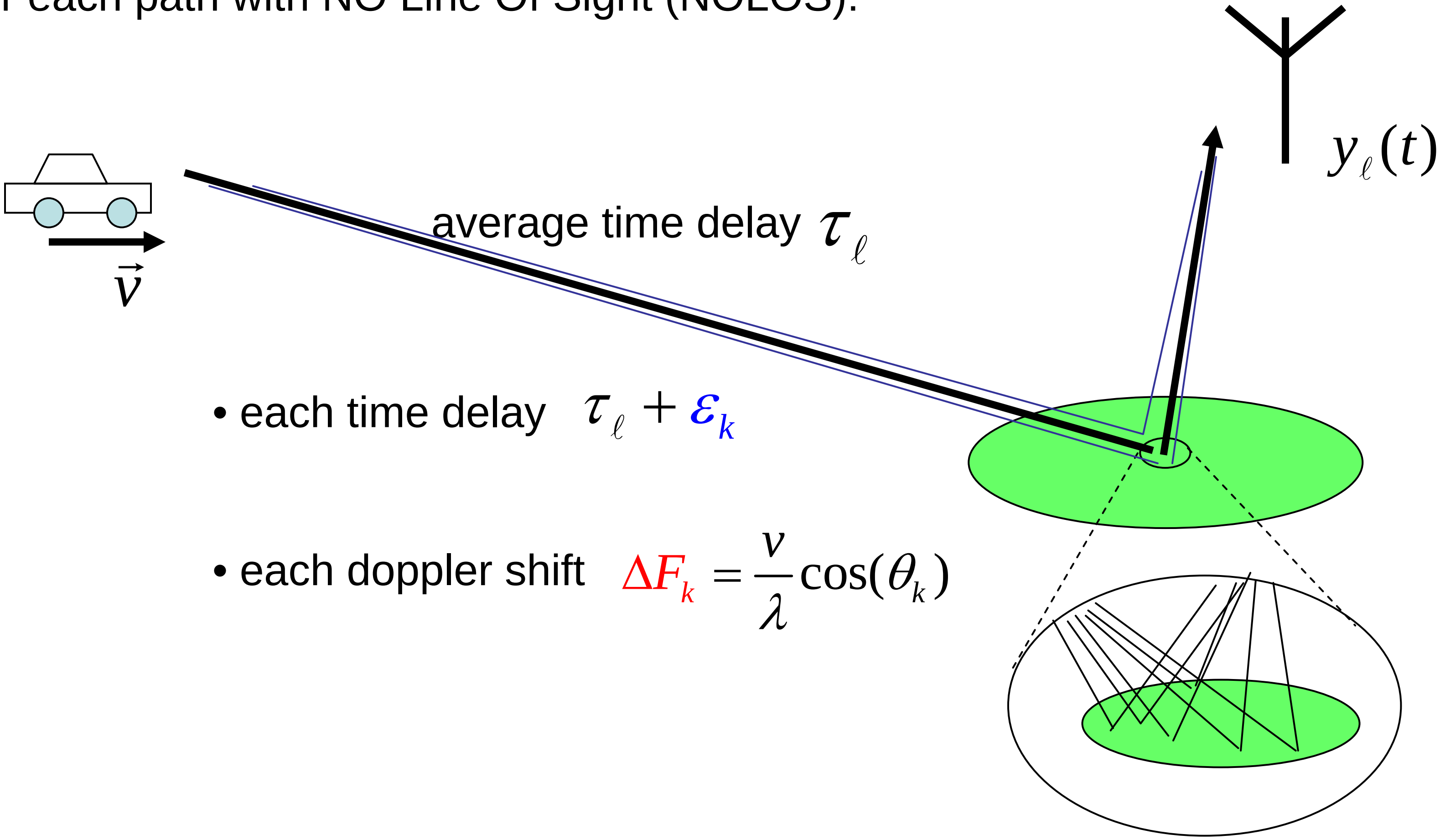
2.1 Statistical Models of Fading Channels

each reflector has several paths all with different time delays and doppler shifts

Several Reflectors:



For each path with NO Line Of Sight (NOLOS):



- each time delay $\tau_\ell + \varepsilon_k$
- each doppler shift $\Delta F_k = \frac{v}{\lambda}\cos(\theta_k)$

$$y_\ell(t) = \mathrm{Re}\left\{\left(\sum_k a_k e^{j2\pi(F_c+\Delta F_k)(t-\tau_\ell-\varepsilon_k)} x(t-\tau_\ell-\varepsilon_k)\right)\right\}$$

Some mathematical manipulation …

$$y(t) = \mathrm{Re}\left\{ \left(\sum_k a_k e^{j2\pi \Delta F_k t} e^{-j2\pi (F_c + \Delta F_k)\tau_k} x(t-\tau_k) \right) e^{j2\pi F_c t} \right\}$$

$$= \mathrm{Re}\left\{ r(t) e^{j2\pi F_c t} \right\} = r_I(t)\cos(2\pi F_c t) - r_Q(t)\sin(2\pi F_c t)$$

In Phase and Quadrature Components

$$r(t) = r_I(t) + j r_Q(t) \cong \sum_k a_k e^{j2\pi \Delta F_k t} e^{-j2\pi (F_c + \Delta F_k)(\tau_\ell + \varepsilon_k)} x(t-\tau_\ell)$$

Assume $x(t) \cong x(t-\varepsilon_k)$

… leading to this:

$$\boxed{r(t) = c_\ell(t) x(t-\tau_\ell)}$$

with $$\boxed{c_\ell(t) = \sum_k a_k e^{j2\pi \Delta F_k t} e^{-j2\pi (F_c + \Delta F_k)(\tau_\ell + \varepsilon_k)}}$$ **random, time varying**

Statistical Model for the time varying coefficients

$$c_\ell(t) = \sum_{k=1}^{M} a_k e^{j2\pi \frac{v}{\lambda}\cos\theta_k t} e^{-j2\pi(F_c + \frac{v}{\lambda}\cos\theta_k)(\tau_\ell + \varepsilon_k)}$$

random

random

By the CLT $c_\ell(t)$ is gaussian with:

$E\{c_\ell(t)\} = 0$ since θ_k random uniformly distributed in $[0, 2\pi]$

$$E\left\{c_\ell(t) c_\ell^*(t+\Delta t)\right\} = \sum_{k=1}^{M} E\{|a_k|^2\} e^{-j2\pi \frac{v}{\lambda}\cos\theta_k \Delta t}$$

Assume $E\{|a_k|^2\} = \frac{P_\ell}{M}$

Then

$$\boxed{\begin{aligned} E\left\{c_\ell(t) c_\ell^*(t+\Delta t)\right\} &= P_\ell \frac{1}{M} \sum_{k=1}^{M} e^{-j2\pi \frac{v}{\lambda}\cos\theta_k \Delta t} = P_\ell E\left\{ e^{-j2\pi \frac{v}{\lambda}\cos\theta \Delta t} \right\} \\ &= P_\ell \frac{1}{2\pi} \int_0^{2\pi} e^{-j2\pi \frac{v}{\lambda}\cos\theta \Delta t} d\theta = P_\ell J_0(2\pi F_D \Delta t) \end{aligned}}$$

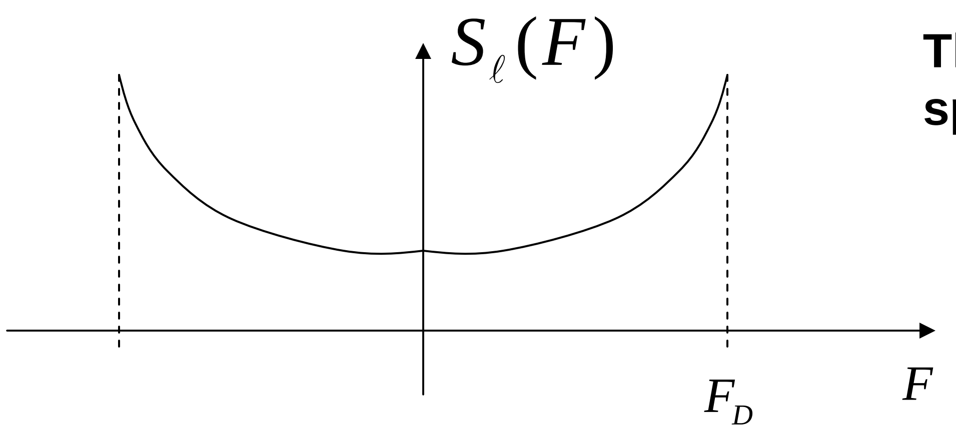
Each coefficient $c_\ell(t)$ is complex, gaussian, WSS with autocorrelation

$$E\left\{c_\ell(t)c_\ell^*(t+\Delta t)\right\} = P_\ell J_0(2\pi F_D \Delta t)$$

And PSD

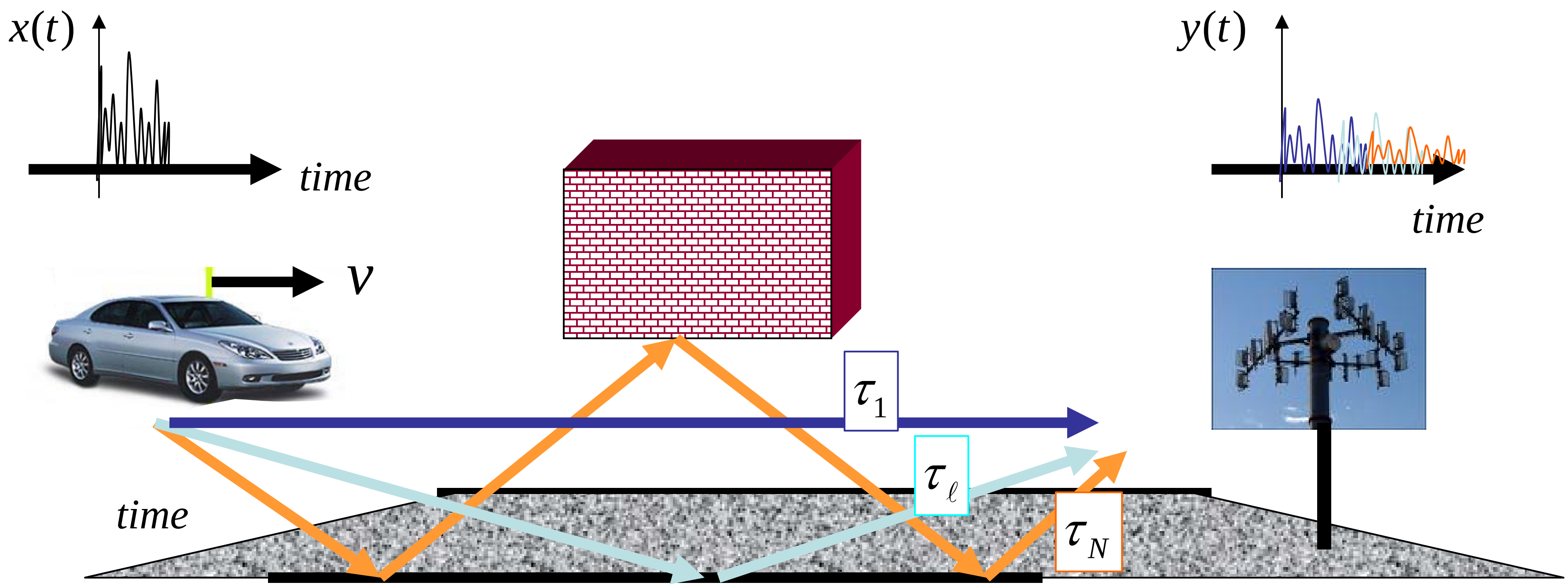
$$S_\ell(F) = \begin{cases} \dfrac{2P_\ell}{\pi F_D} \dfrac{1}{\sqrt{1-(F/F_D)^2}} & \text{if} \quad |F| < F_D \\ 0 & \text{otherwise} \end{cases}$$

with F_D **maximum Doppler frequency.**

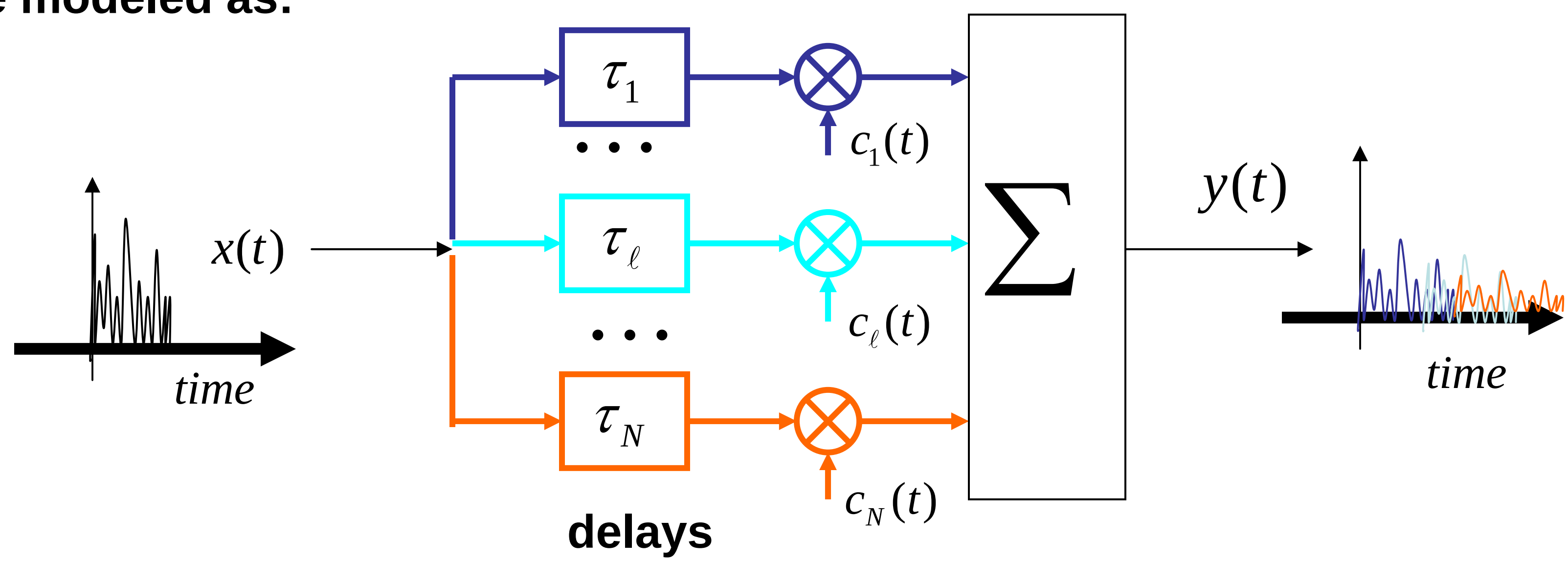


This is called *Jakes* spectrum.

Bottom Line. This:



… can be modeled as:



For each path

$$c_\ell(t) = \sqrt{P_\ell}\, g(t)$$

• **time invariant**

• **from power distribution**

• **unit power**

• **time varying (from autocorrelation)**

Parameters for a Multipath Channel (No Line of Sight):

Time delays:	$\begin{bmatrix} \tau_1 & \tau_2 & \cdots & \tau_L \end{bmatrix}$	sec
Power Attenuations:	$\begin{bmatrix} P_1 & P_2 & \cdots & P_L \end{bmatrix}$	dB
Doppler Shift:	F_D	Hz

Non Line of Sight (NOLOS) and Line of Sight (LOS) Fading Channels

1. **Rayleigh (No Line of Sight).** $E\{c_\ell(t)\} = 0$

Specified by:

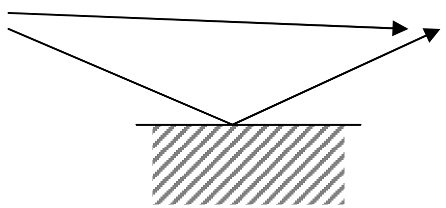
Time delays $T = [\tau_1, \tau_2, ..., \tau_N]$

Power distribution $P = [P_1, P_2, ..., P_N]$

Maximum Doppler F_D

2. Ricean (Line of Sight) $E\{c_\ell(t)\} \neq 0$

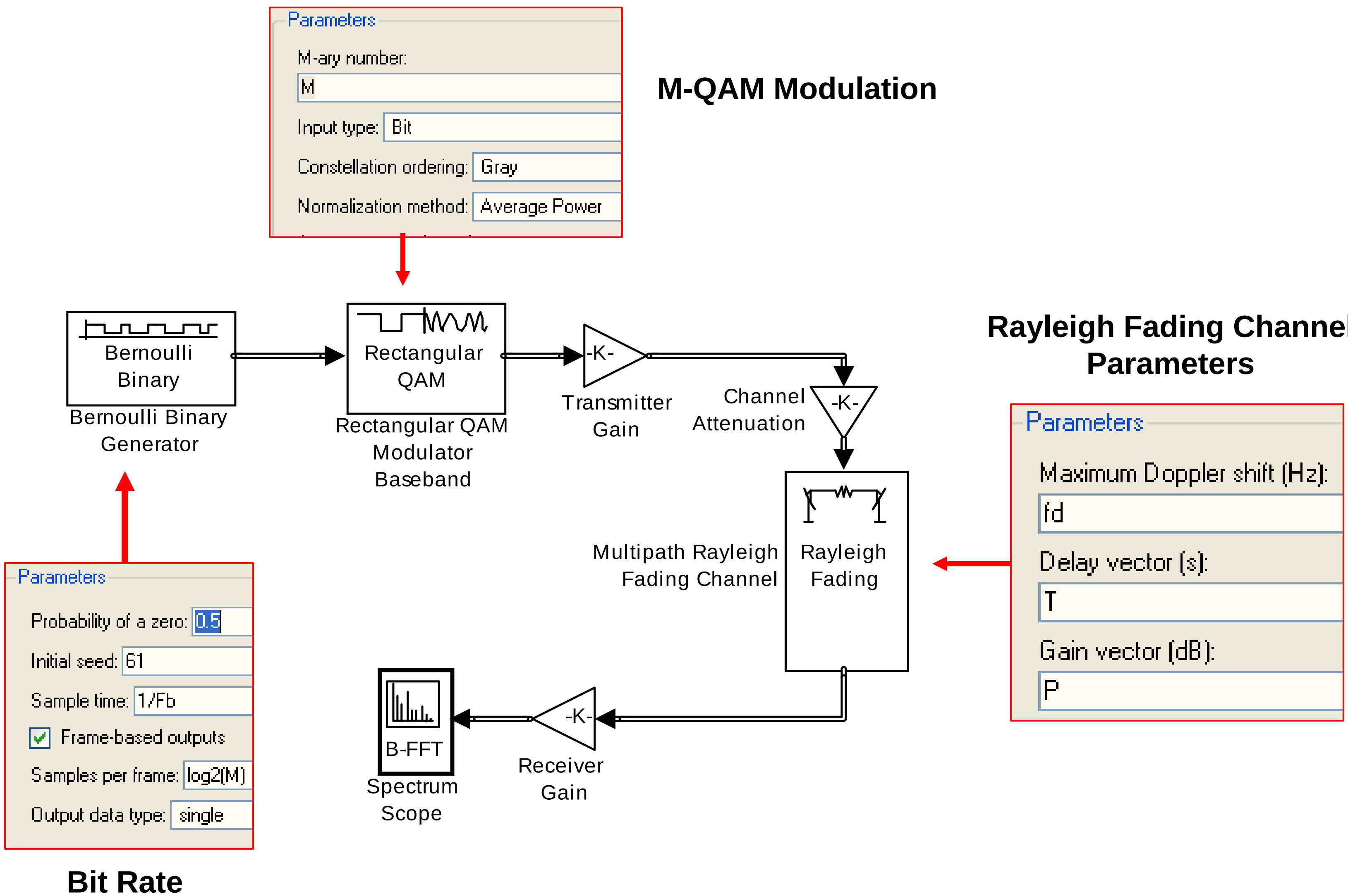
Same as Rayleigh, plus Ricean Factor K



Power through LOS $P_{LOS} = \frac{K}{1+K} P_{Total}$

Power through NOLOS $P_{NOLOS} = \frac{1}{1+K} P_{Total}$

Simulink Example



Set Numerical Values:

Recall the Doppler Frequency:

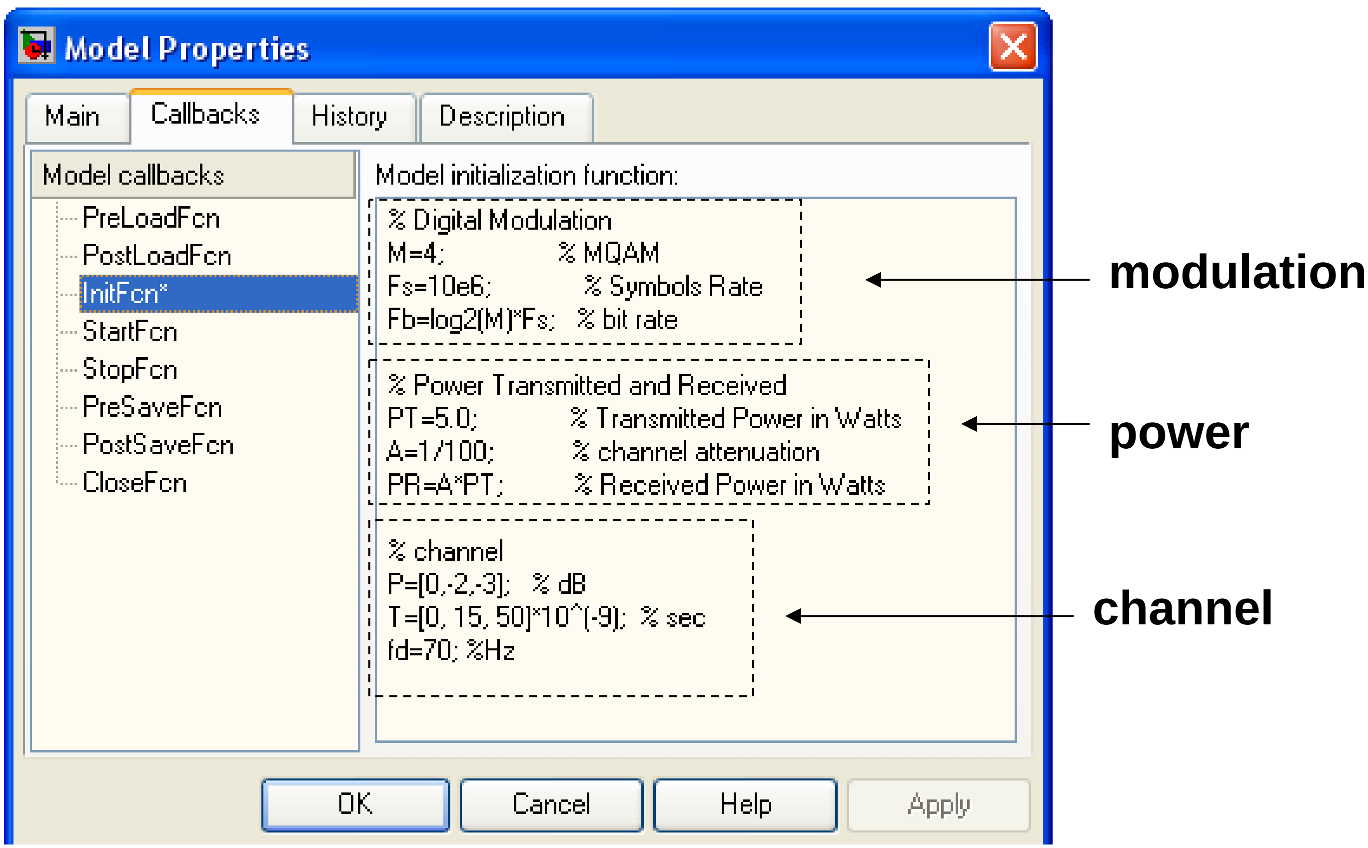
velocity

carrier freq.

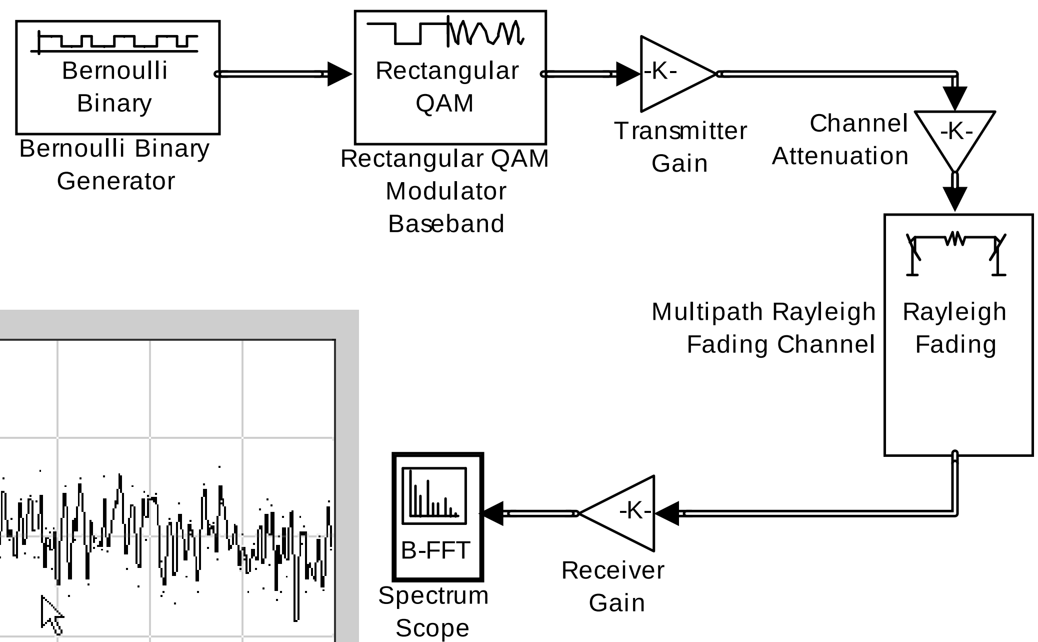
$$F_D = \frac{v}{c} F_C$$

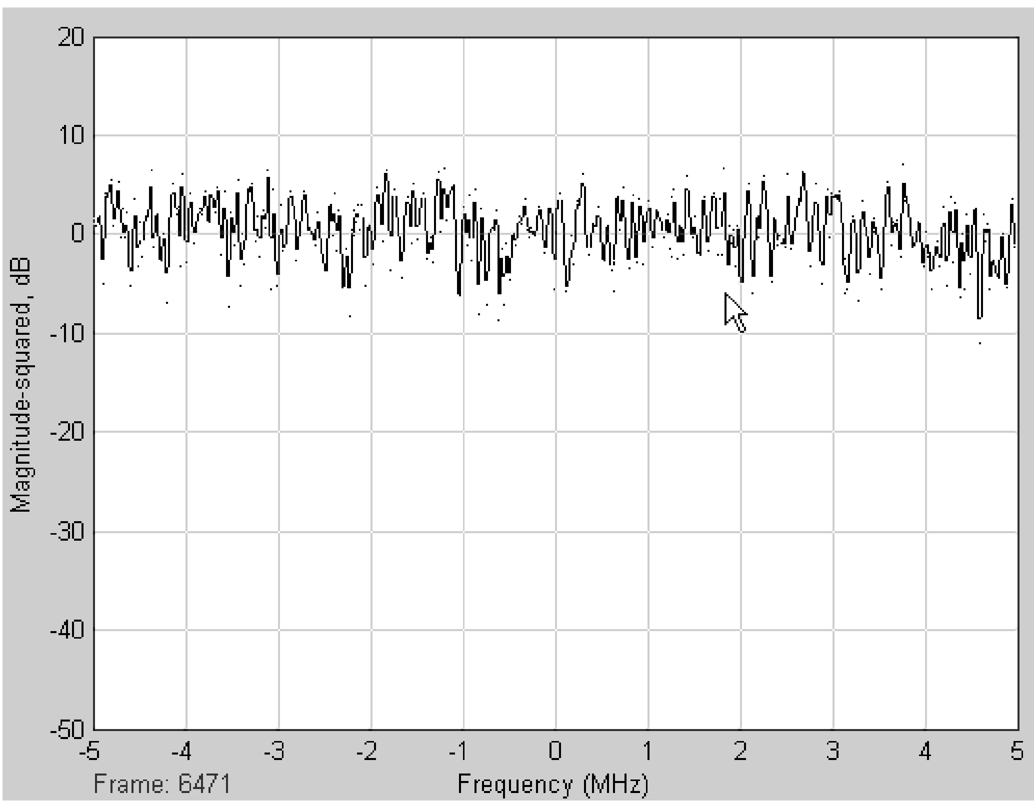
$3 \times 10^8 \, m/\sec$

Easy to show that: $\boxed{(F_D)_{Hz} \approx (v)_{km/h} (F_C)_{GHz}}$



Typical Received Power Spectrum





$-F_S/2$ $F_S/2$

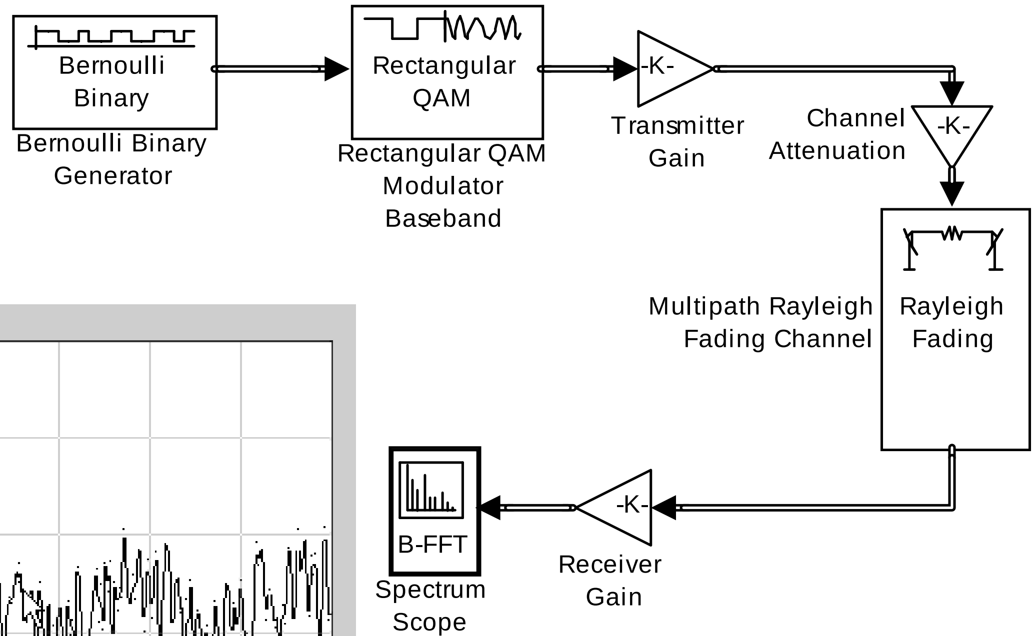
Doppler freq: $F_D = 70Hz$

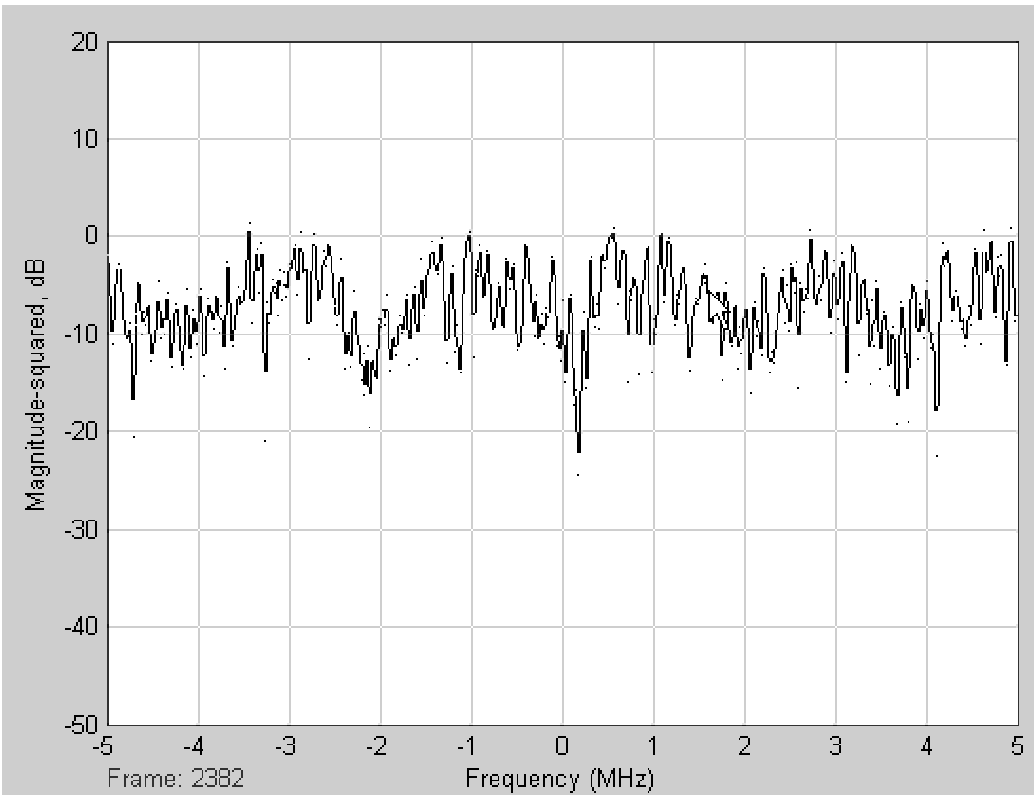
Symbol rate: $F_S = 10MHz$

Bit rate: $F_b = 20MHz$

Bits/Symbol: 2 (QPSK)

Similar Example, different parameters





Doppler freq: $F_D = 2Hz$

Symbol rate: $F_S = 10MHz$

Bit rate: $F_b = 20MHz$

Bits/Symbol: 2 (QPSK)

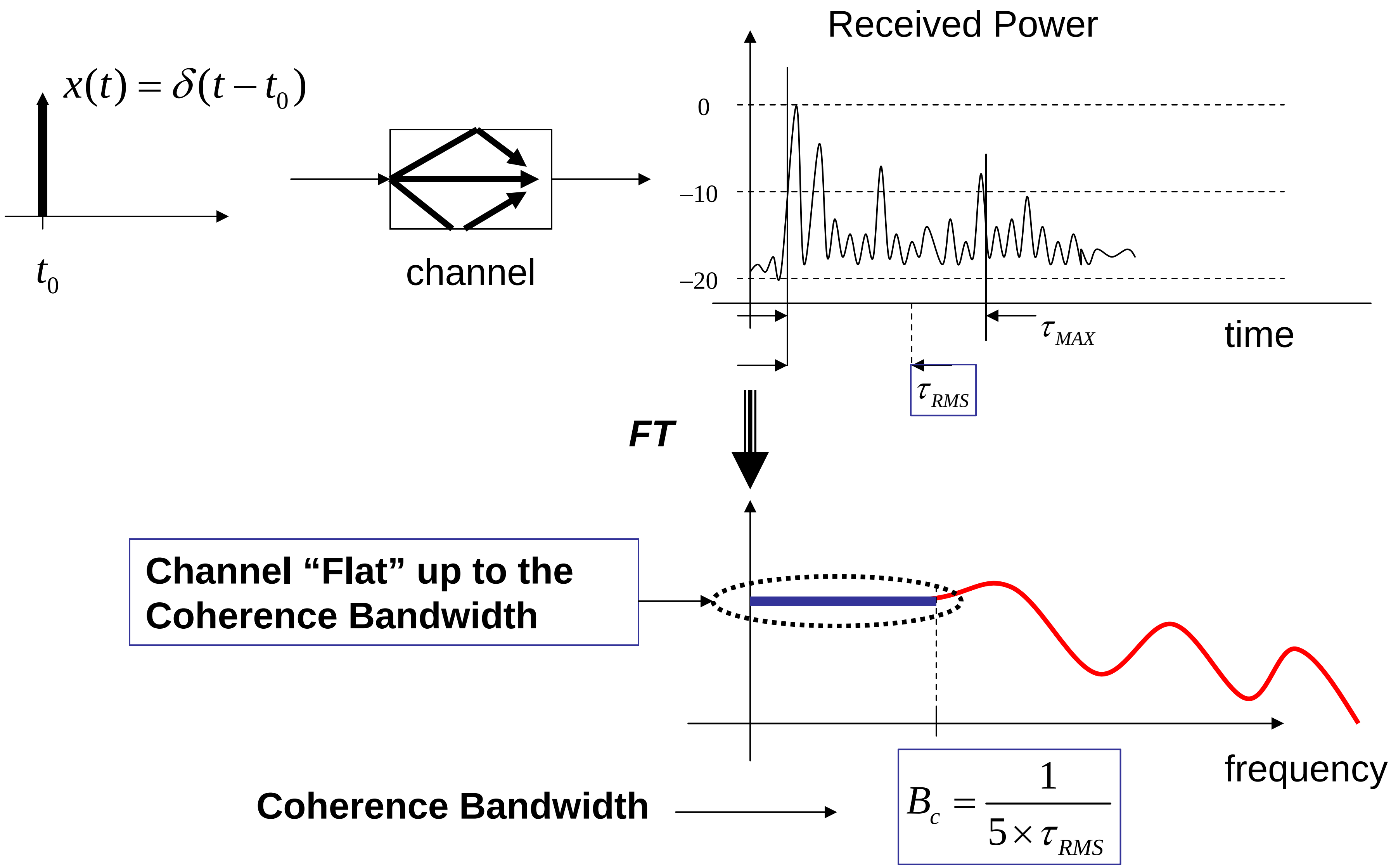
$-F_S/2$ $F_S/2$

Channel Parameterization

1. Time Spread and Frequency Coherence Bandwidth
2. Flat Fading vs Frequency Selective Fading
3. Doppler Frequency Spread and Time Coherence
4. Slow Fading vs Fast Fading

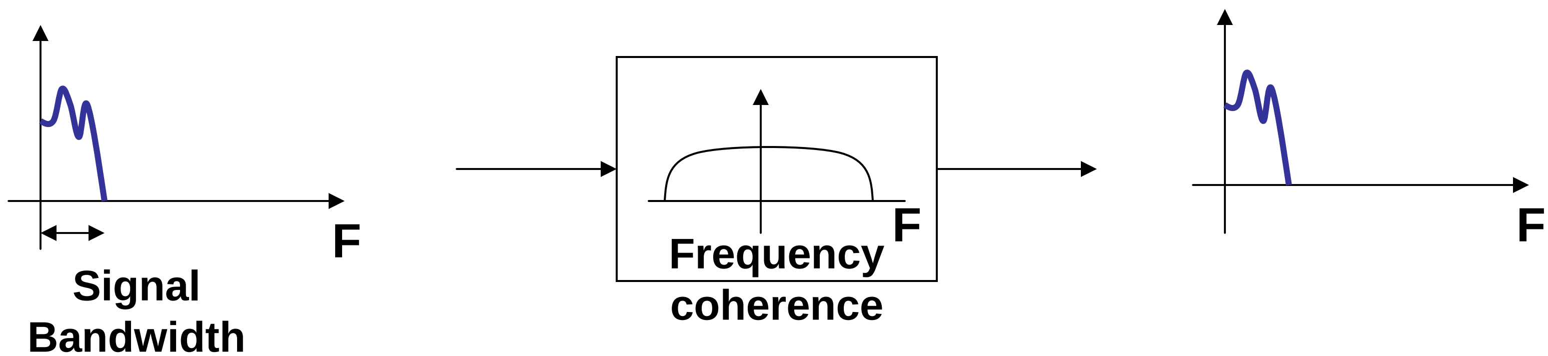
1. Time Spread and Frequency Coherence Bandwidth

Recall that the Channel Spreads in Time (due to Multipath):



2. Flat Fading vs Frequency Selective Fading

- Based on Channel Time Spread: $$B_c = \frac{1}{5 \times \tau_{RMS}}$$



Flat Fading — **Just attenuation, no distortion**

Signal Bandwidth $<$ / $>$ **Frequency Coherence**

Frequency Selective Fading — **Distortion!!!**

Example: Flat Fading

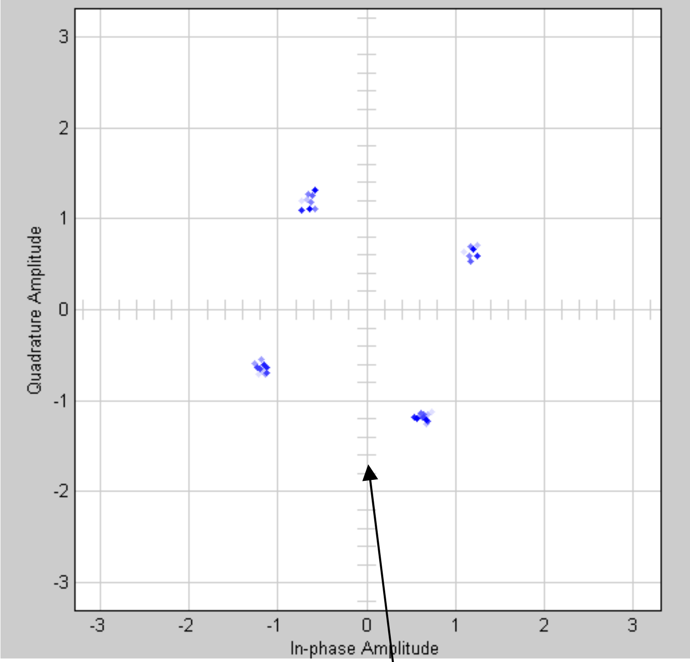
Channel : **Delays T=[0 10e-6 15e-6] sec**

Power P=[0, -3, -8] dB

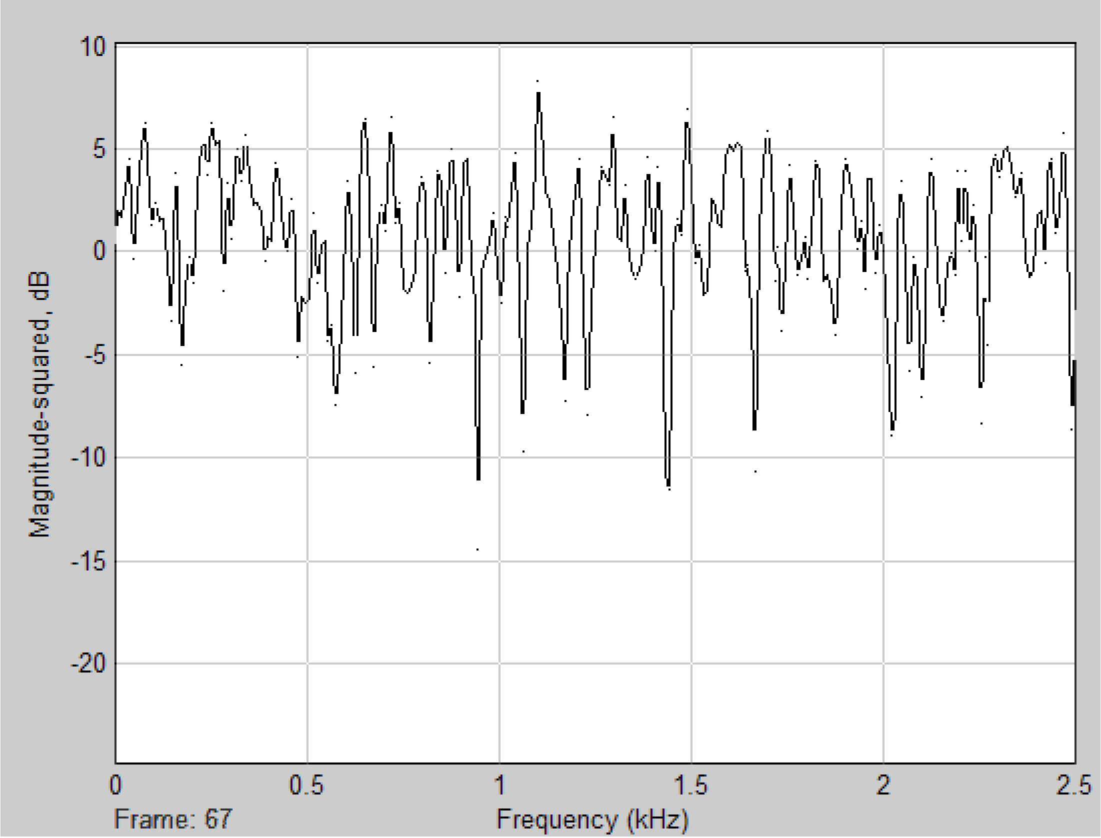
Symbol Rate Fs=10kHz

Doppler Fd=0.1Hz

Modulation QPSK



Very low Inter Symbol Interference (ISI)



Spectrum: fairly uniform

Example: Frequency Selective Fading

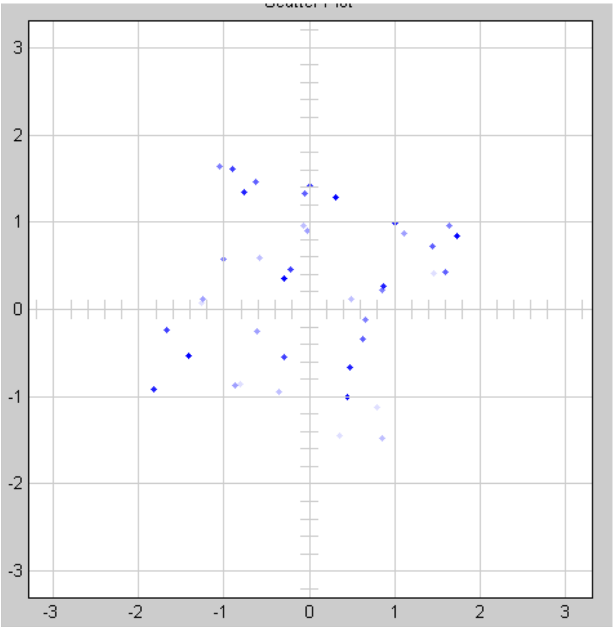
Channel : Delays T=[0 10e-6 15e-6] sec

Power P=[0, -3, -8] dB

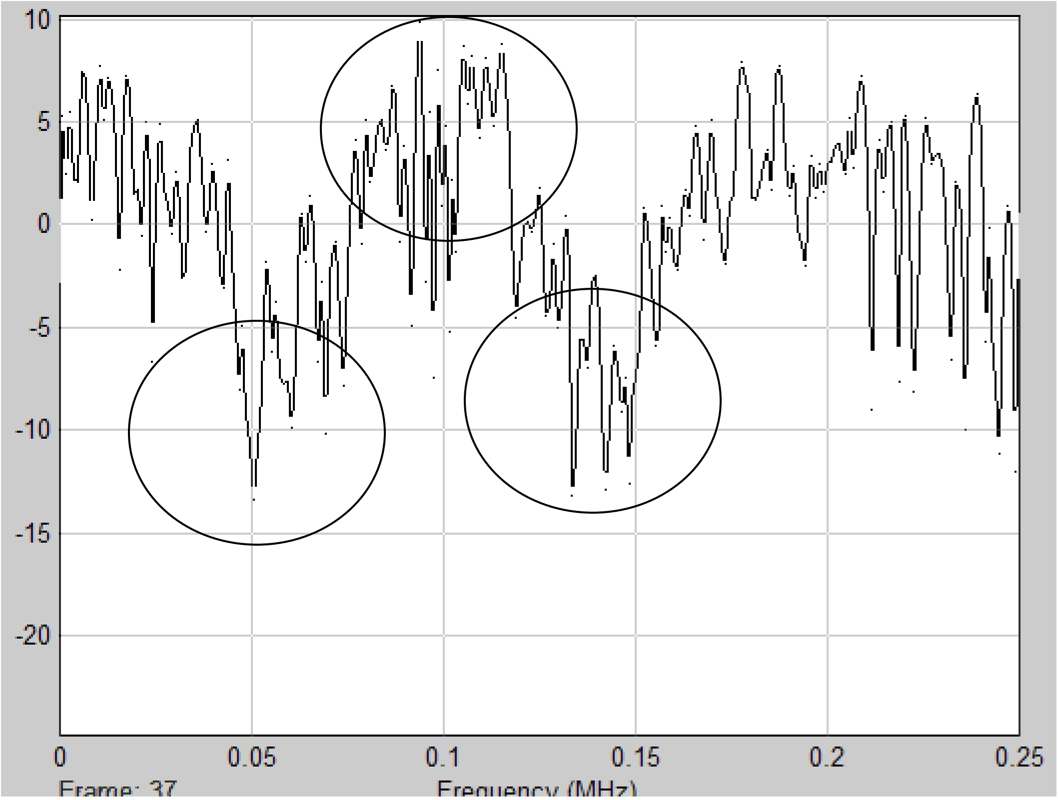
Symbol Rate Fs=1MHz

Doppler Fd=0.1Hz

Modulation QPSK



Very high ISI

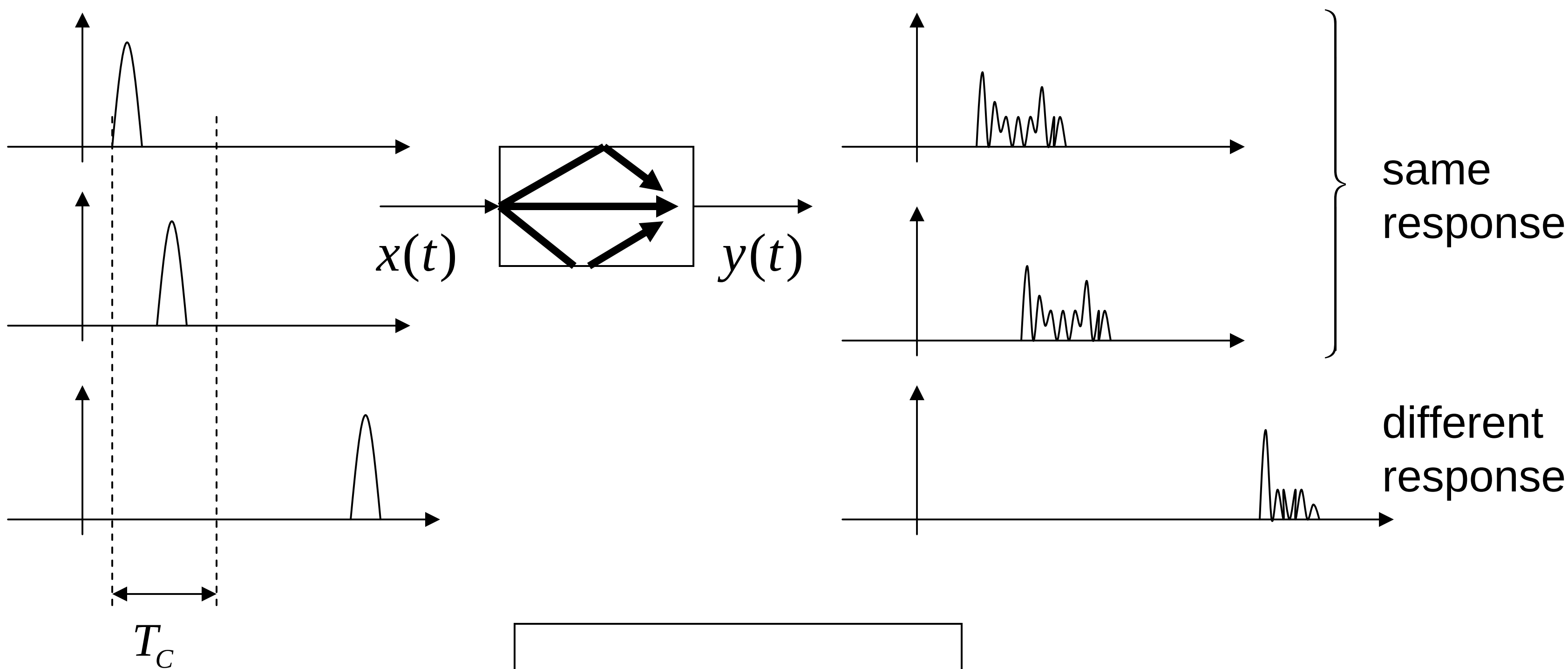


Spectrum with deep variations

3. Doppler Frequency Spread and Time Coherence

$$y(t) = \sum_{\ell=1}^{M} c_{\ell}(t)\,x(t-\tau_{\ell})$$

$$c_{\ell}(t) \cong c_{\ell}(t+\Delta t) \quad \textbf{if} \quad |\Delta t| \le T_C < 1/F_D$$



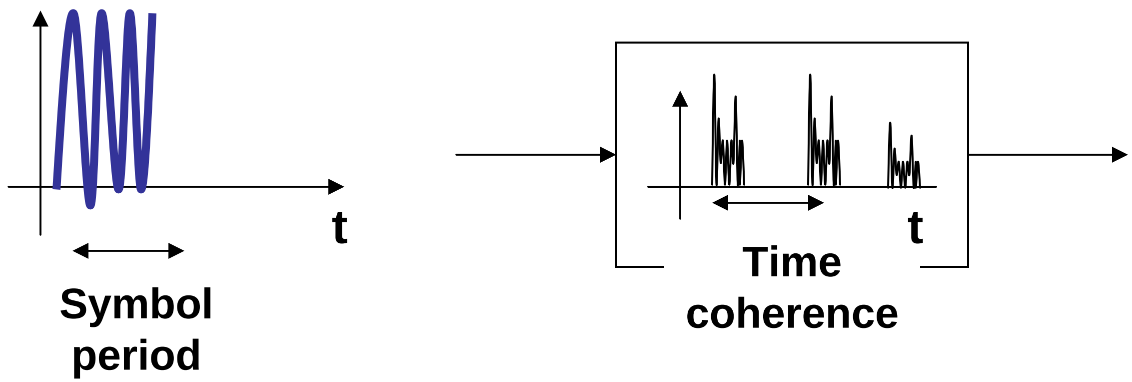
Coherence Time:

$$\boxed{T_C \approx \frac{9}{16\pi F_D}}$$

Max Doppler

4. Slow Fading vs Fast Fading

• Based on Doppler Spread Delay and Time Coherence: $$T_C \approx \frac{9}{16\pi F_D}$$

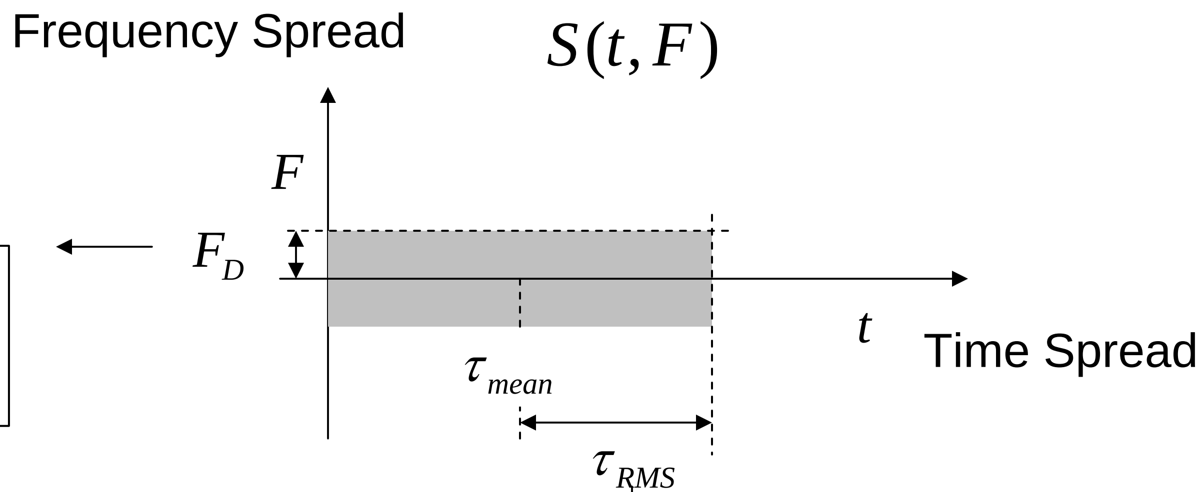


Slow Fading — **Channel almost time invariant**

Symbol period $\lt$ / $\gt$ **Time Coherence**

Fast Fading — **Channel quickly changing**

Summary of Time/Frequency spread of the channel



Time Coherence

$$\boxed{T_C \approx \frac{9}{16\pi F_D}}$$

Frequency Coherence

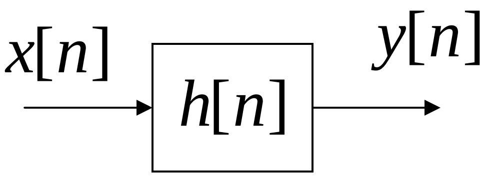
$$\boxed{B_c = \frac{1}{5 \times \tau_{RMS}}}$$

Channel Estimation from Data

1. Recall Impulse Response Identification from Correlation
2. Estimation of Time Spread and Doppler Shift
3. Simulink/Matlab Example
4. Stanford University Interim (SUI) Channel Models

Estimation of Channel Characteristics from Input - Output data.

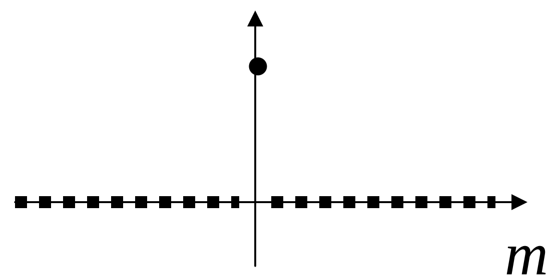
1. For Linear Time Invariant (LTI) systems:



$$y[n] = h[n] * x[n] = \sum_{m=-\infty}^{+\infty} h[k]x[n-k]$$

Excite the system with white noise and unit variance

$$R_{xx}[m] = E\left\{x[n]x^*[n-m]\right\} = \delta[m]$$

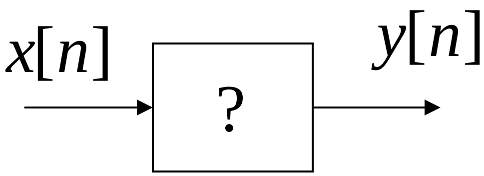


and compute the crosscorrelation between input and output

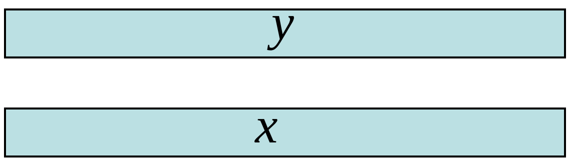
$$R_{yx}[m] = E\left\{y[n]x^*[n-m]\right\}$$

$$= \sum_{k=-\infty}^{+\infty} h[k]E\left\{x[n-k]x^*[n-m]\right\} = \sum_{k=-\infty}^{+\infty} h[k]\delta[m-k] = h[m]$$

In matlab:



1. Get data (same length for simplicity):



2. Compute crosscorrelation between input and output:

h=xcorr(x,y);

If **x[n]** is white noise, **h[n]** is the impulse response.

2. For a Linear Time Varying Channel:

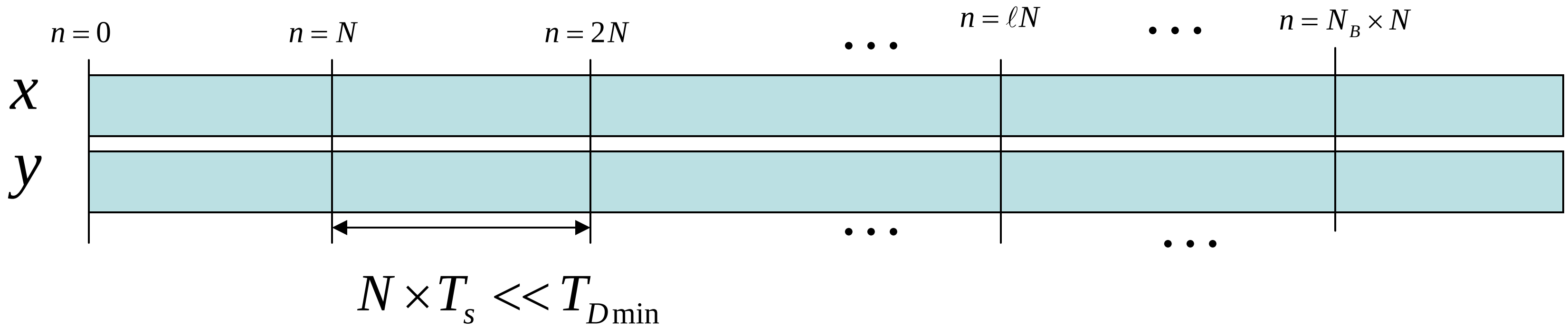
$$x[n] \longrightarrow \boxed{h[m,n]} \longrightarrow y[n]$$

$$y[n] = \sum_{k=-\infty}^{+\infty} h[k,n]x[n-k]$$

Known:

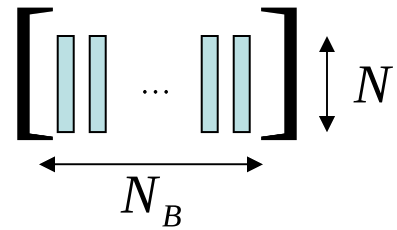
1. Sampling frequency F_s
2. Upper bound on max Doppler Frequency $F_{D\max}$

1. Collect Data and partition in blocks of length $N << \frac{F_s}{F_{D\max}}$:




```
xN=reshape(x,N,length(x)/N);
yN=reshape(x,N,length(y)/N);
```

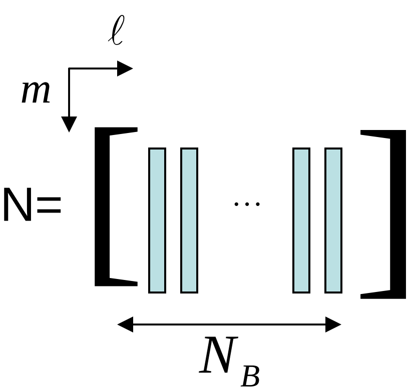
xN,yN=



2. Estimate impulse response in each block

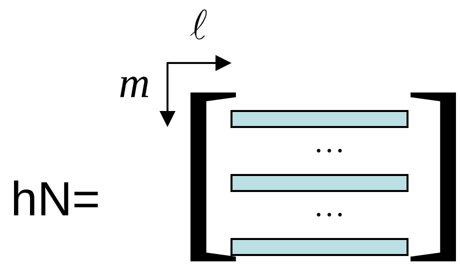
```
hN=xcorr(xN,yN);
```

hN=



$h[m,n]\big|_{n=\ell N}$

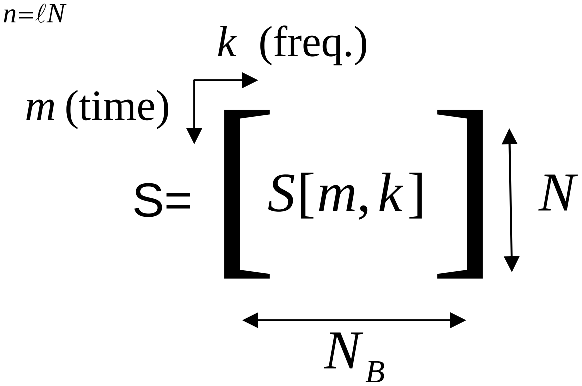
3. Compute Power Spectrum on each row, to determine time variability of the channel (If the channel is Time Invariant all columns of hN are the same):



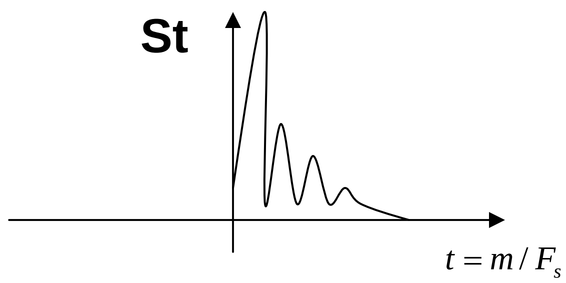
$$h[m,n]\big|_{n=\ell N}$$

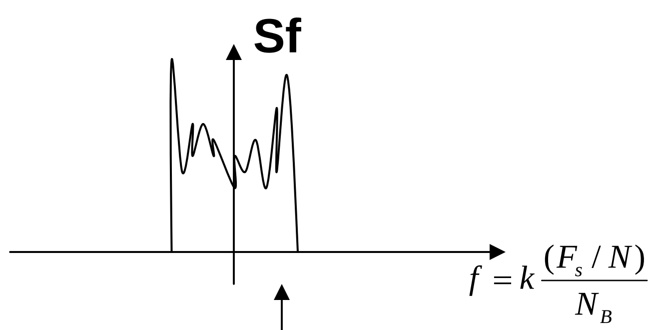
```
S=fft(hN');
S=S'.*conj(S');
```

$\Rightarrow$



4. Take the sum over rows for Doppler Spread and sum over columns for Time Spread (fftshift each vector to have “zero” term (sec or Hz) in the middle





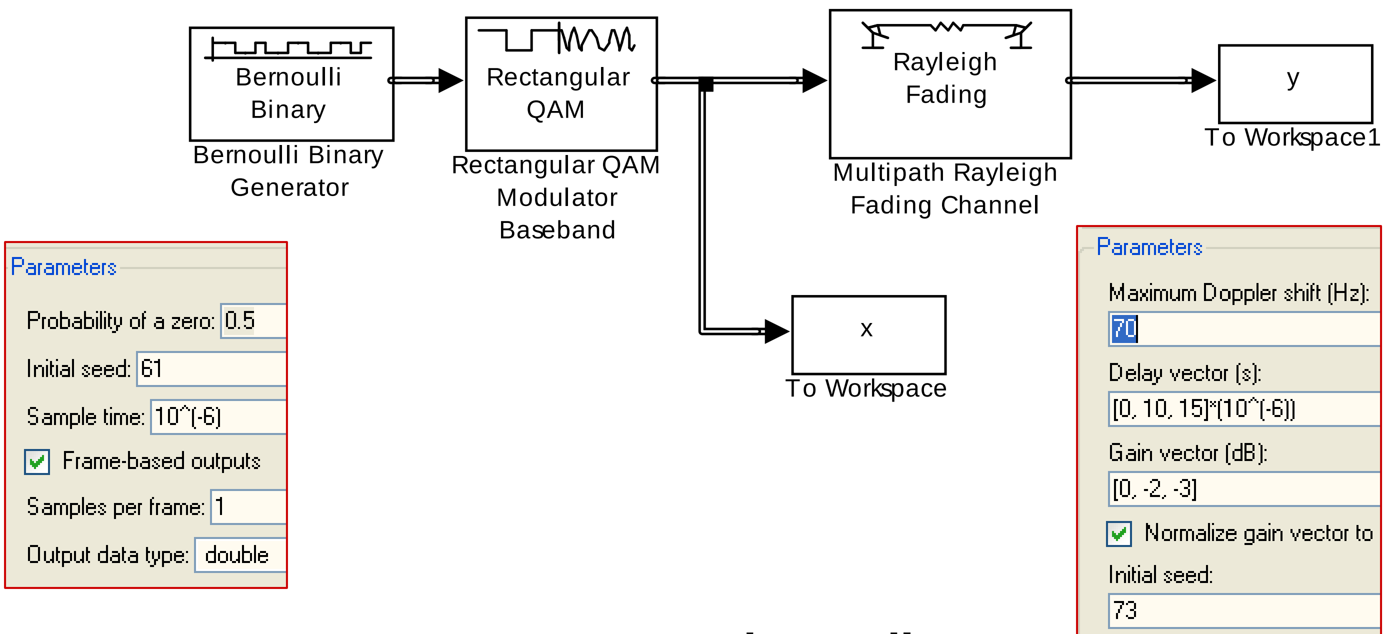
Time Resolution: $\Delta t = 1 / F_S$

Frequency Resolution: $\Delta F = \dfrac{F_S}{N \times N_B} = \dfrac{1}{\text{total data length(sec)}} Hz$

Therefore if we want to a resolution in the doppler spread of (say) 1Hz, we need to collect at least 1 sec of data.

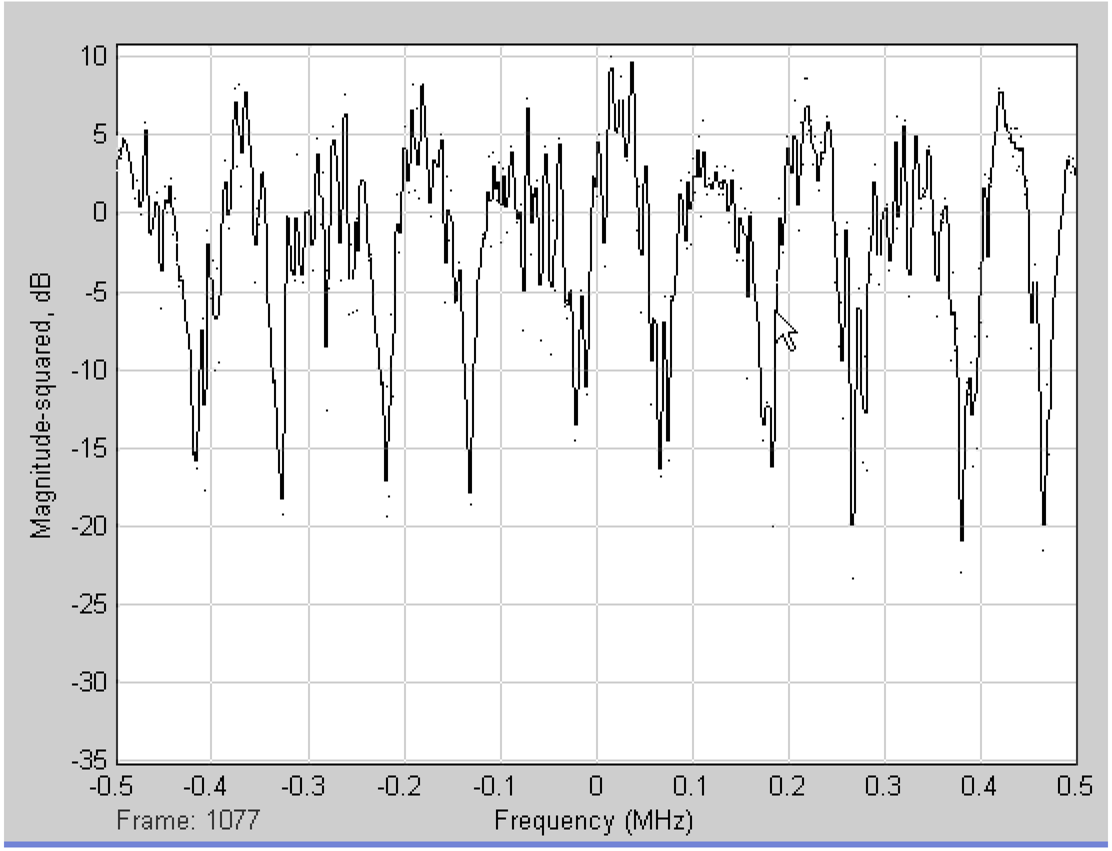
Example:

```
% channel
Fs=10^6;                      % sampling freq. In Hz
P=[0,-2,-3];                  % attenuations in dB
T=[0, 10, 15]*10^(-6);        % time delays in sec
fd=70;                        %doppler shift in Hz
```



test_scattering.mdl

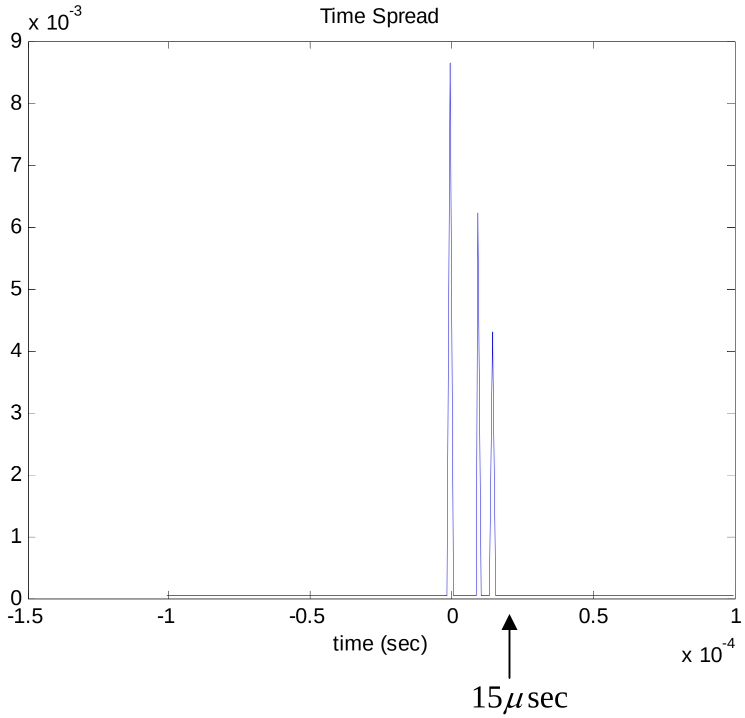
Channel Freq. Response:



Tmax=10^(-4) sec; **[St, Sf,t,f]=scattering(x,y,Fs,Tmax, FDmax);**

FDmax=150Hz;

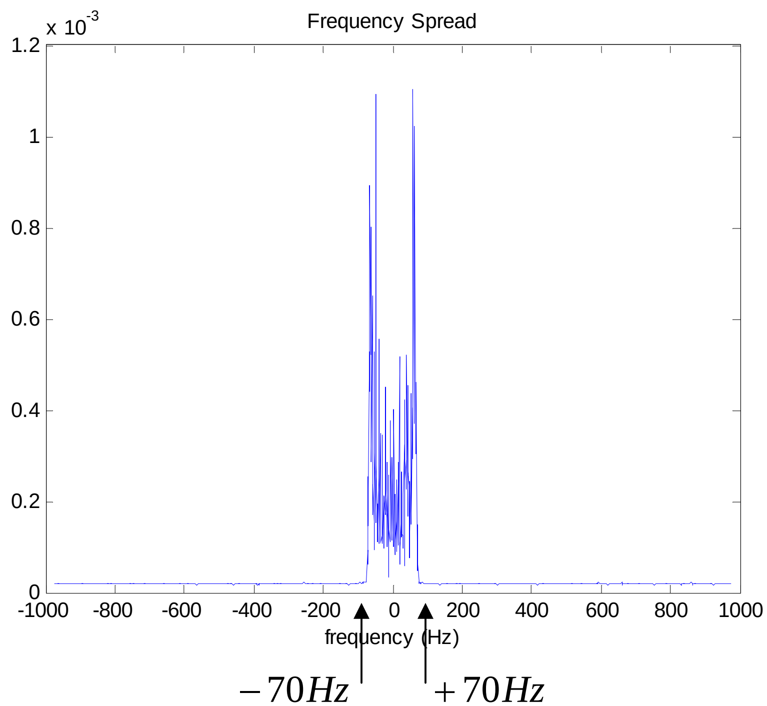
St(t)



Time Spread

Frequency Spread

Sf(f)



Stanford University Interim (SUI) Channel Models

Extension of Work done at AT&T Wireless and Erceg etal.

Three terrain types:

- Category A: Hilly/Moderate to Heavy Tree density;
- Category B: Hilly/ Light Tree density or Flat/Moderate to Heavy Tree density
- Category C: Flat/Light Tree density

Six different Scenarios (SUI-1 – SUI-6).

Found in

IEEE 802.16.3c-01/29r4, “Channel Models for Wireless Applications,” http://wirelessman.org/tg3/contrib/802163c-01_29r4.pdf

V. Erceg etal, “An Empirical Based Path Loss Model for Wireless Channels in Suburban Environments,” IEEE Selected Areas in Communications, Vol 17, no 7, July 1999

Lab 3

In this project you want to identify the time and frequency spreads of a channel. Refer to the simulink model **Lab3.mdl** for the setup.

You know that the mobile channel you are trying to model has a maximum doppler frequency not exceeding 50Hz and a maximum time spread smaller than 20 microseconds. In order to have a clear picture of the channel spread in time and frequency we want a time resolution of 1 microsecond and a frequency resolution of 1Hz;

1. Based on the time and frequency resolutions, determine a suitable symbol rate of the transmitted sequence and an adequate data length in time;
2. With the above parameters, run the model **Lab3.mdl** to collect the transmitted and received data. Use the program **scattering.m** to estimate the time and frequency spreads of the channel. Compare the result with the parameters of the channel in the simulink block;
3. Just to compare, try different values of symbol rate and data length and see how the resolutions in time and frequency are affected.

4. Multi Carrier Modulation and OFDM

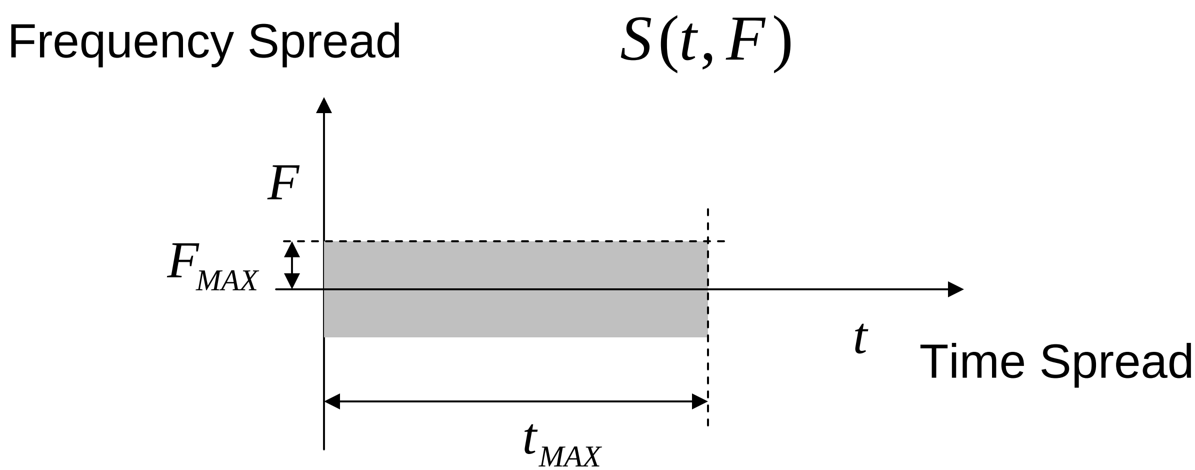
1. Single Carrier and Multi Carrier Modulation
2. Orthogonal Frequency Division Multiplexing (OFDM)
3. Example: basics of IEEE 802.11a (WiFi)

Single Carrier and Multi Carrier Modulation

1. Transmission of Data through Frequency Selective and Time Varying Channels
2. Single Carrier Modulation in Flat Fading Channels
3. Single Carrier Modulation in Frequency Selective Channels
4. Simulink Example of Single Carrier Modulation
5. The Multi Carrier Approach

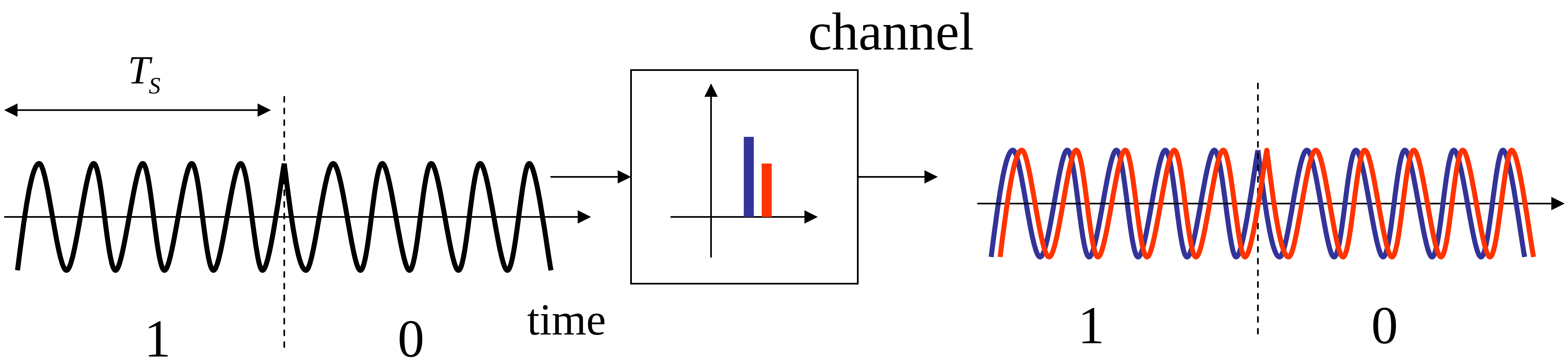
1. Transmission of Data Through Frequency Selective Time Varying Channels

We have seen a wireless channel is characterized by *time spread* and *frequency spread.*



2. Single Carrier Modulation in Flat Fading Channels:

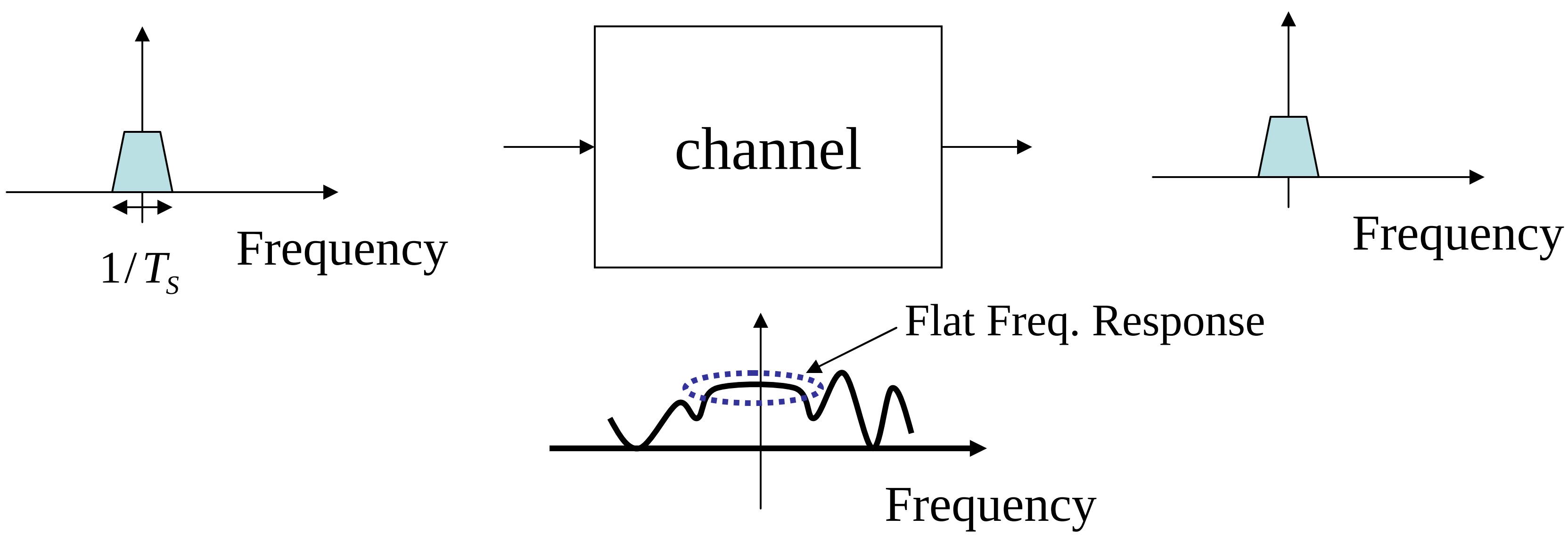
• if *symbol duration* >> *time spread* then there is almost no Inter Symbol Interference (ISI).



phase still recognizable

Problem with this: *Low Data Rate!!!*

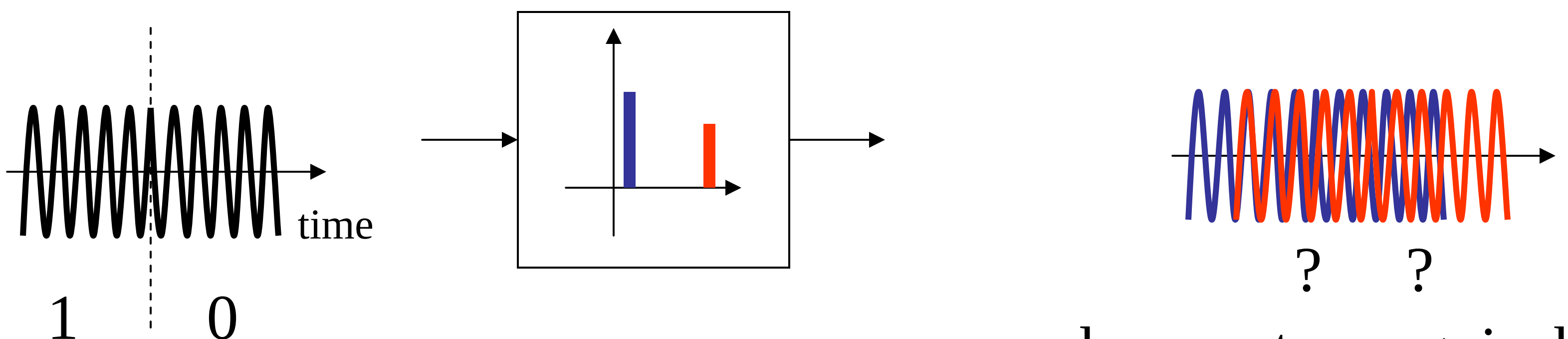
• this corresponds to **Flat Fading**



3. Single Carrier Modulation in Frequency Selective Channels:

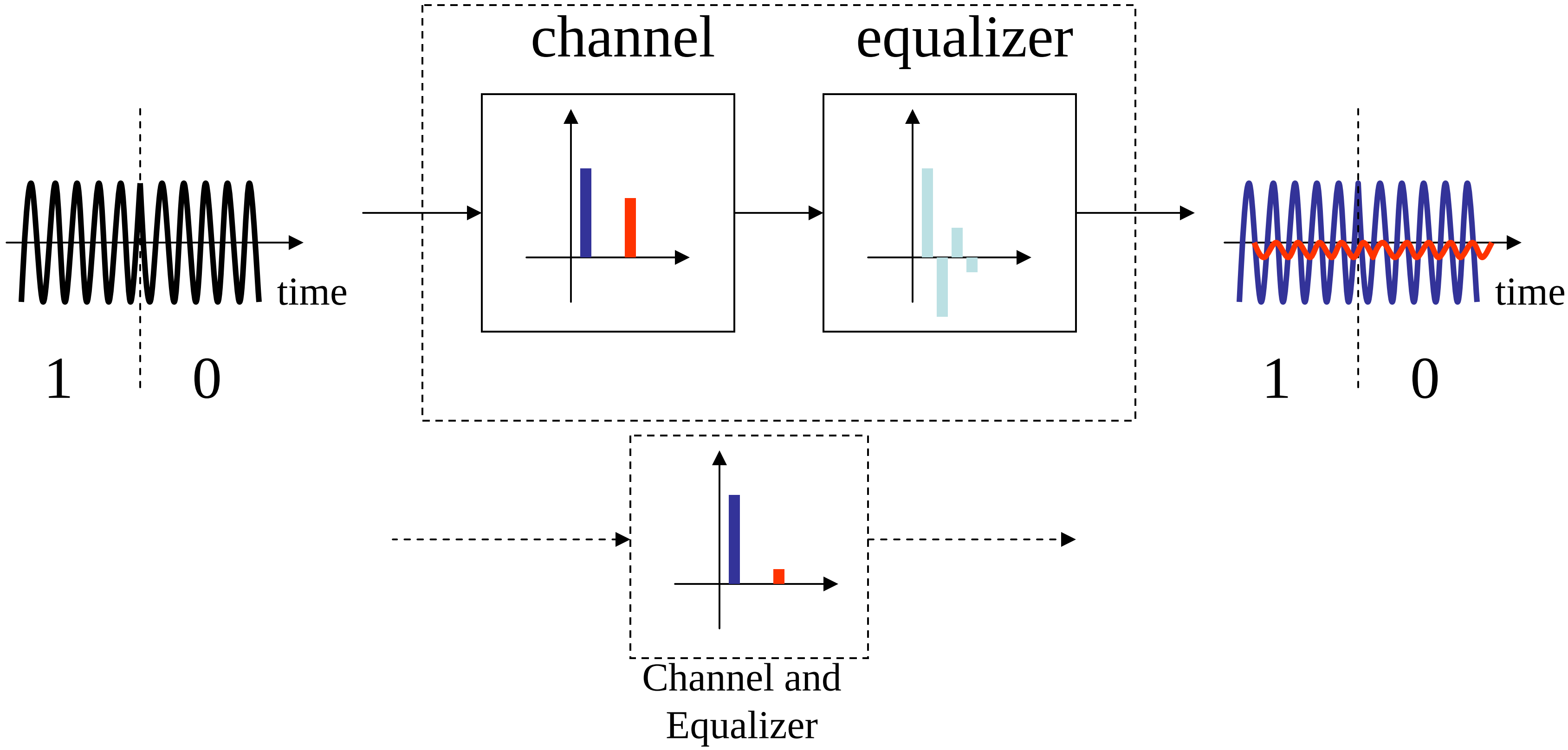
• if *symbol duration ~ time spread* then there is considerable Inter Symbol Interference (ISI).

channel



phase not recognizable

One Solution: *we need equalization*



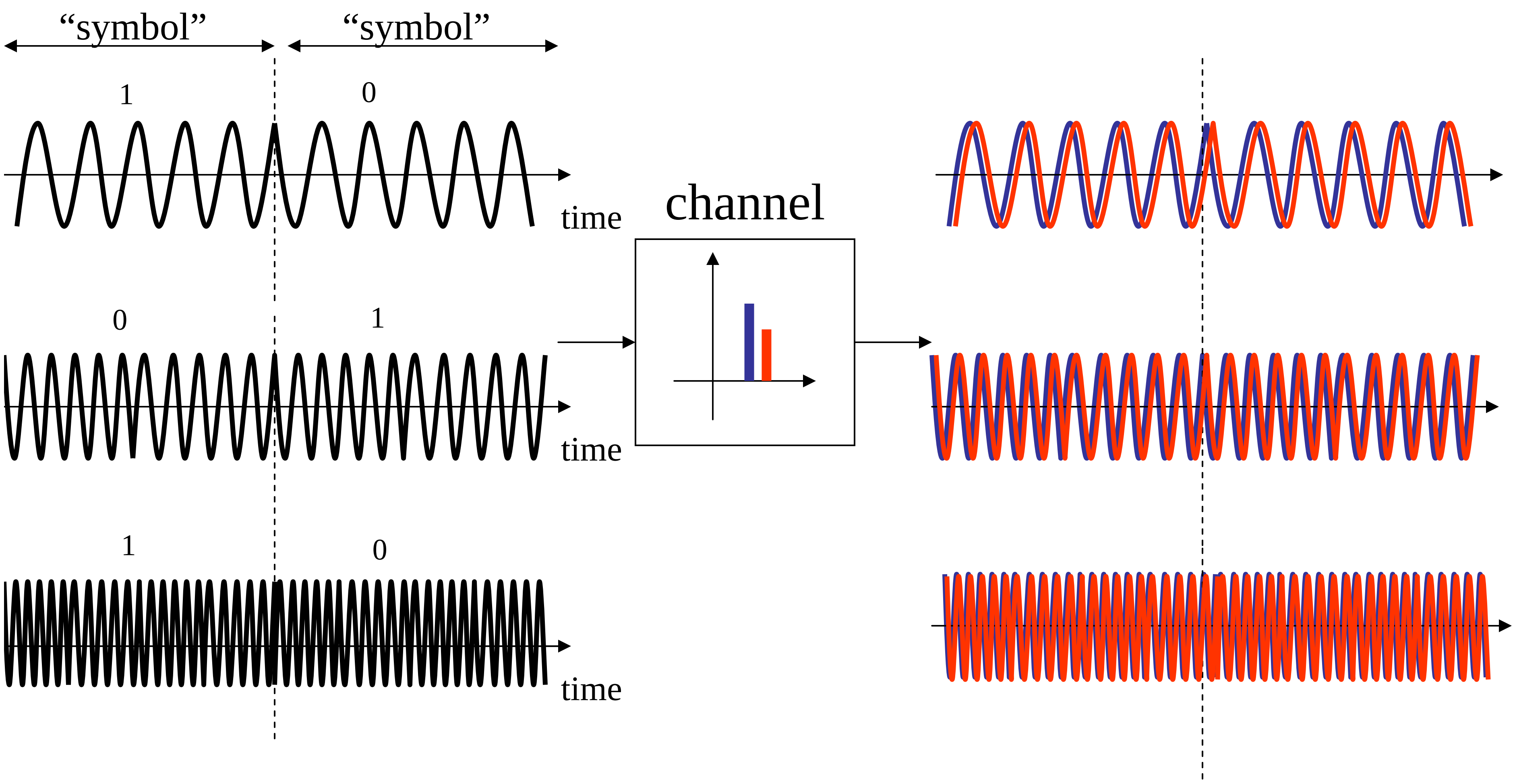
Problems with equalization:

- it might require training data (thus loss of bandwidth)
- if blind, it can be expensive in terms computational effort
- always a problem when the channel is time varying

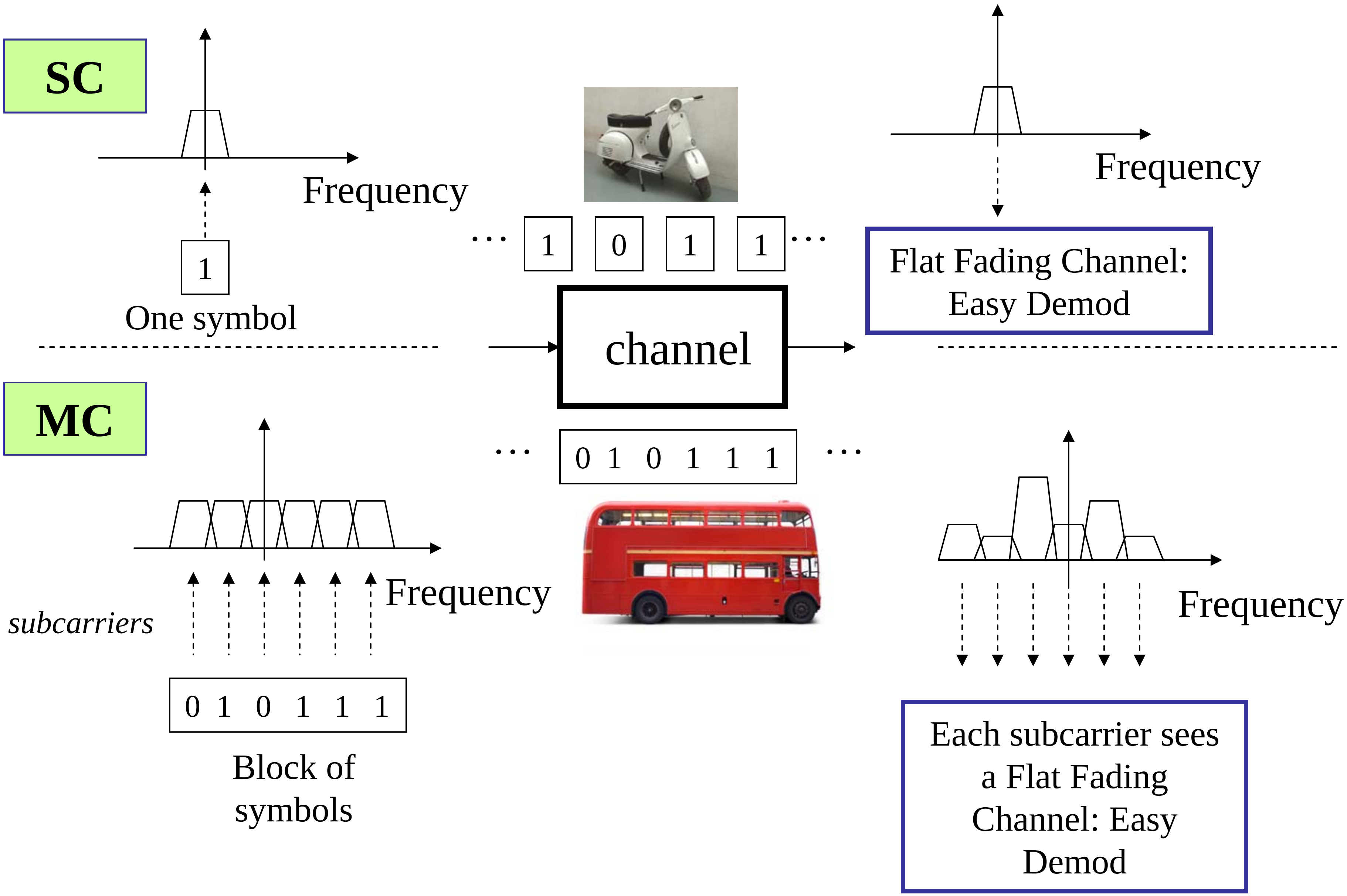
4. The Multi Carrier Approach:

• let *symbol duration* >> *time spread* so there is almost no Inter Symbol Interference (ISI);

• send a **block of data** using a number of carriers (Multi Carrier)



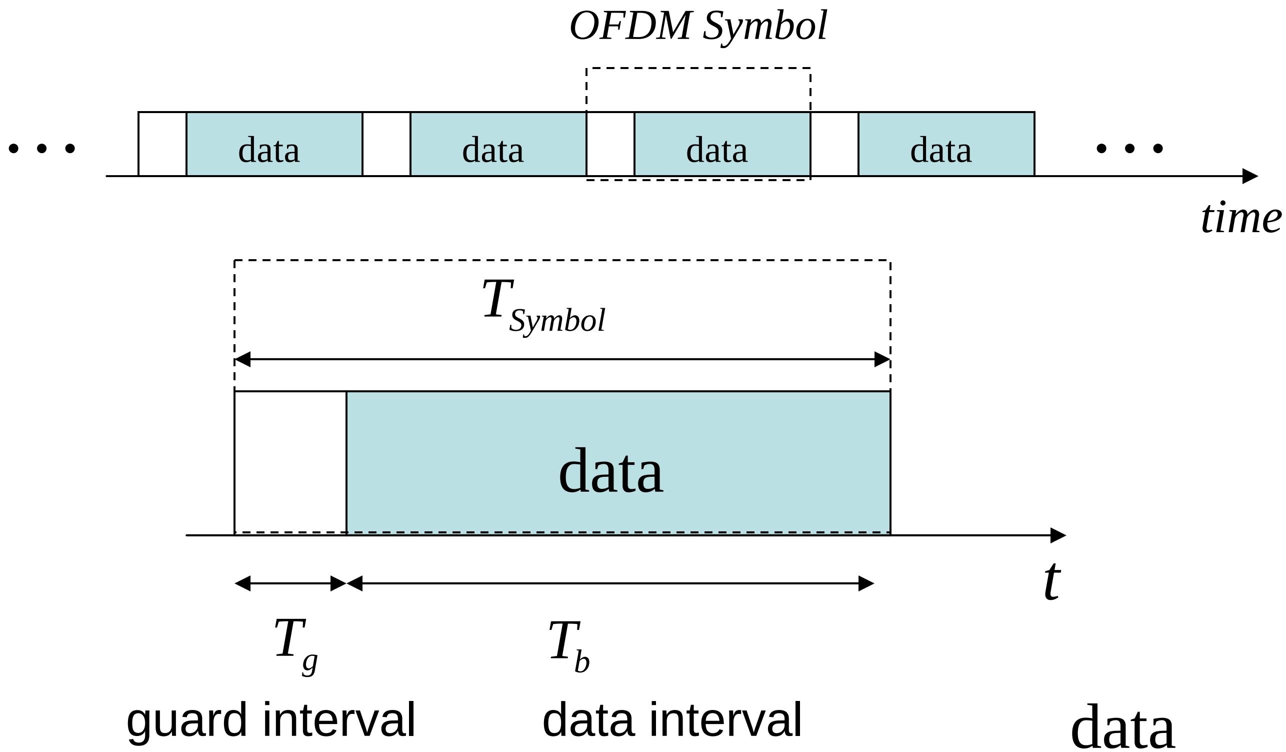
Compare Single Carrier and Multi Carrier Modulation



Orthogonal Frequency Division Multiplexing (OFDM)

1. Basic Structure of Multi Carrier Modulation
2. “Orthogonal” Subcarriers and OFDM
3. Generating the OFDM Symbol using the IFFT

In MC modulation each “MC symbol” is defined on a time interval and it contains a block of data



data

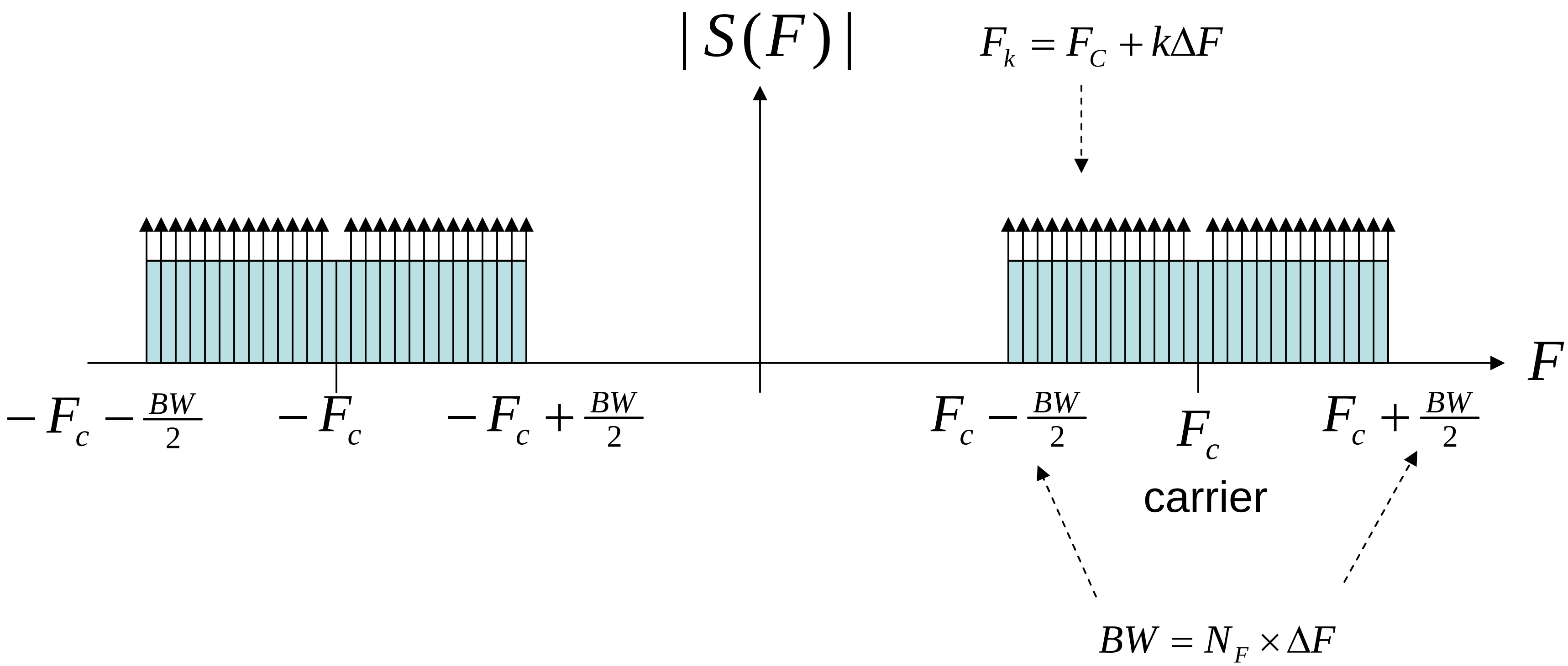
$$s(t) = \mathrm{Re}\left\{ A \sum_{k=-\frac{N_F}{2}}^{\frac{N_F}{2}} c_k e^{j2\pi(F_C + k\Delta F)t} \right\} = \mathrm{Re}\left\{ e^{j2\pi F_C t} A \sum_{k=-\frac{N_F}{2}}^{\frac{N_F}{2}} c_k e^{j2\pi k\Delta F t} \right\}$$

$$0 \le t \le T_{Symbol}$$

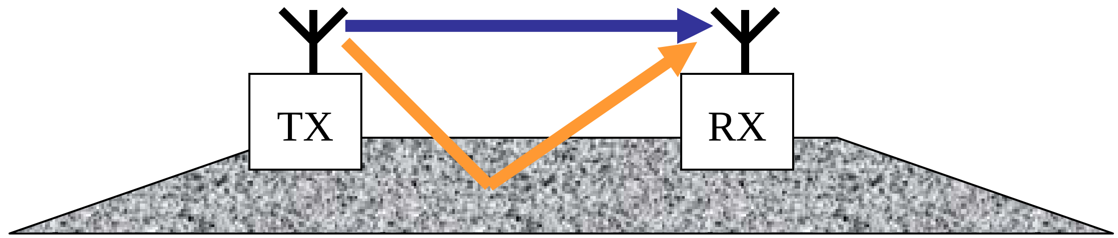
•Each data point c_k is modulated by a subcarrier

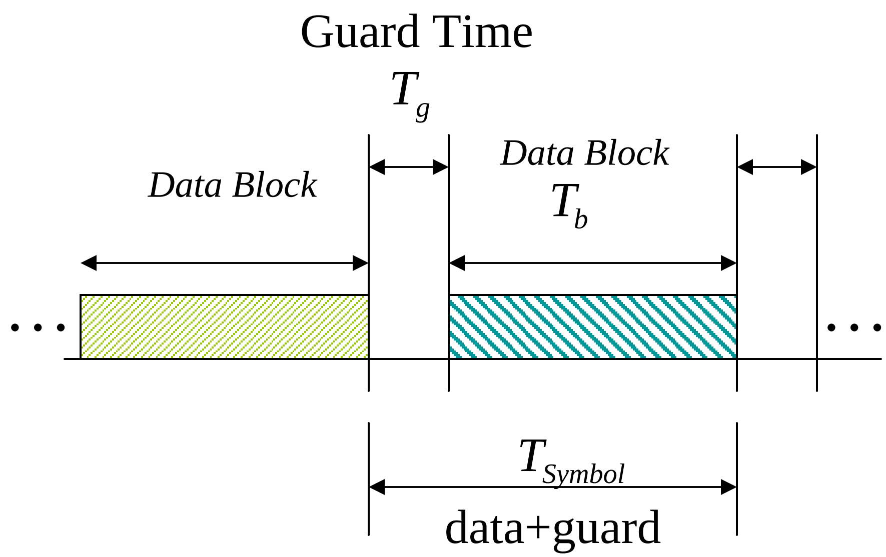
$$F_k = F_C + k\Delta F, \quad k = -\frac{N_F}{2}, \ldots, \frac{N_F}{2}, \quad k \neq 0$$

•Subcarrier $k = 0$ is not used since its magnitude and phase would be influenced by the carrier F_C

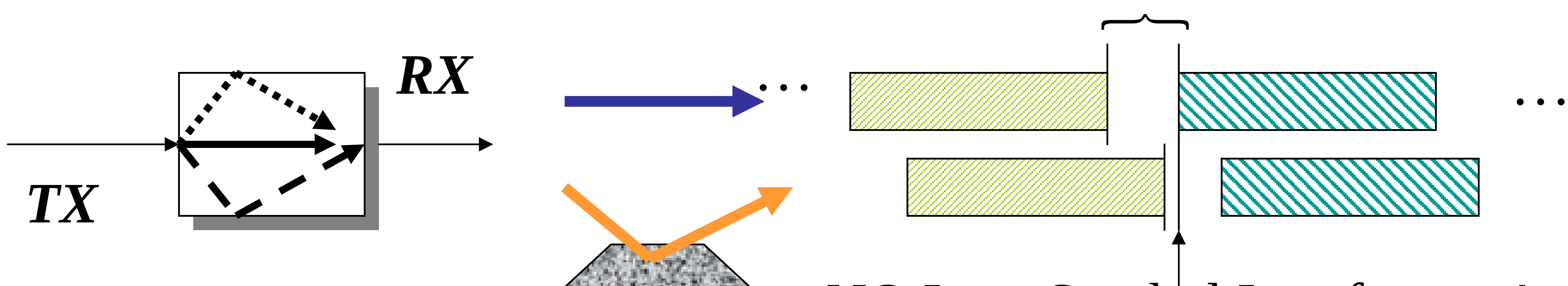


We leave a "guard time" between blocks to allow multipath





the "guard time" is long enough, so the multipath in one block does not affect the next block

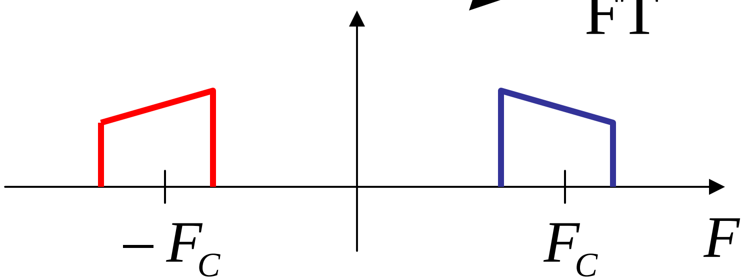


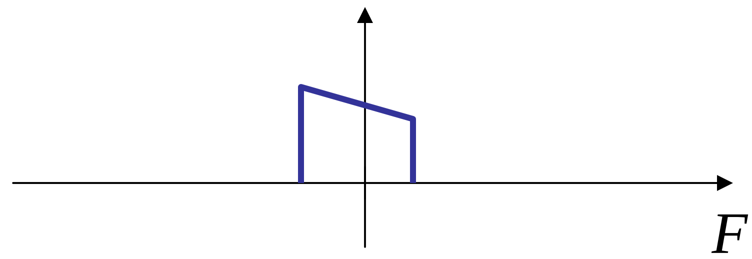
NO Inter Symbol Interference!

Modulated Signal: $s(t) = \mathrm{Re}\left\{e^{j2\pi F_c t} x(t)\right\}$

F_C = carrier frequency

FT





FT

Baseband Complex Signal:

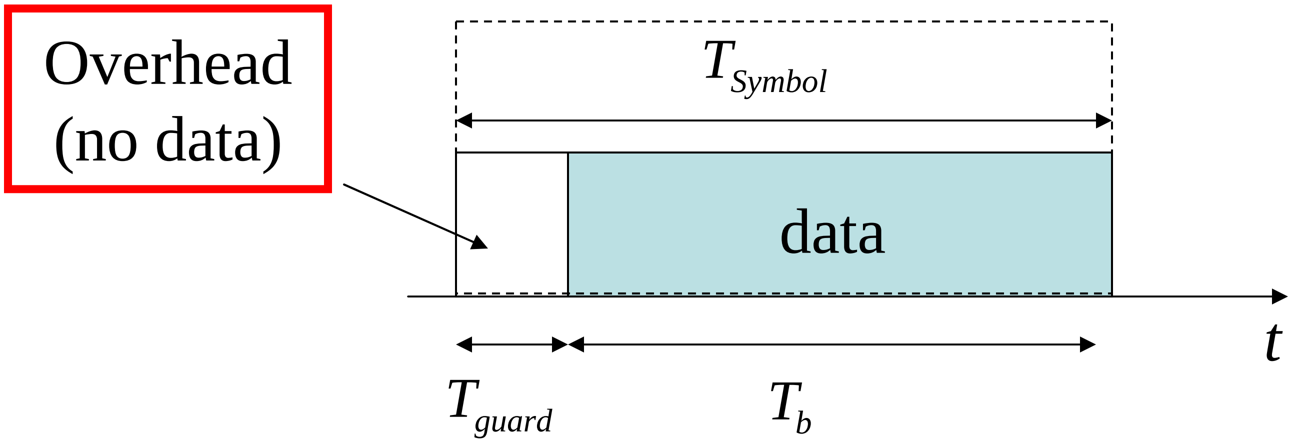
$$x(t) = A \sum_{\substack{k=-\frac{N_F}{2} \\ k \neq 0}}^{k=\frac{N_F}{2}} c_k e^{j2\pi k\Delta F(t-T_g)} \qquad 0 \le t \le T_{Symbol}$$

just a gain

$k\Delta F$ = subcarrier frequency offset

c_k = data

Limitations of OFDM



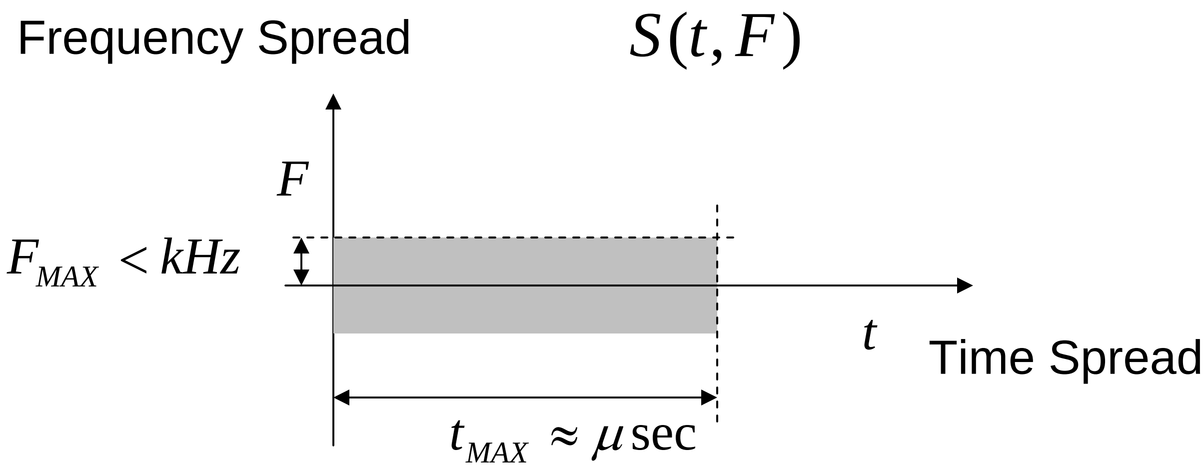
•The Guard Time (or Cyclic Prefix) does not carry data and therefore it represents a loss of power

$$P_{data} = \frac{T_b}{T_b + T_{guard}} P_{symbol} = \frac{1}{1 + \left(\frac{T_{guard}}{T_b} \right)} P_{symbol}$$

•To minimize overhead $\quad T_{guard} << T_b$

ie the longer the data frame the better!

However, as expected, the channel (not the sky!) is the limit.



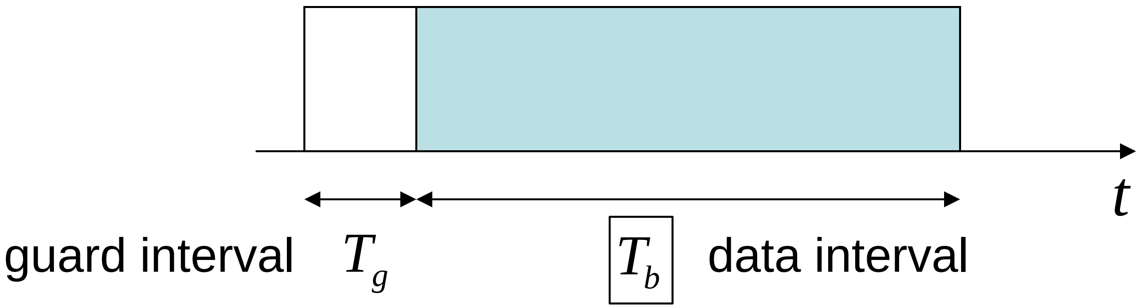
OFDM Symbol Duration: $T_b >> t_{MAX}$ to minimize CP overhead

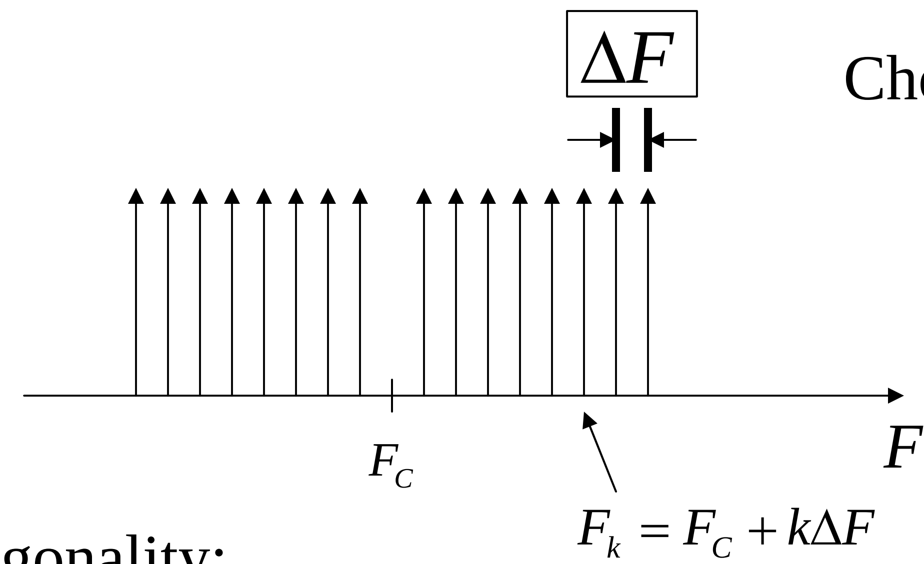
OFDM Freq. Spacing: $\Delta F = \frac{F_S}{N} >> F_{MAX}$ to ensure orthogonality

<u>Are these two compatible?</u>

Since $\frac{F_S}{N} = \frac{1}{NT_S} = \frac{1}{T_b}$ we have: $\underset{\boxed{10^{-6}\sec}}{t_{MAX}} << T_b << \underset{\boxed{10^{-3}\sec}}{1/F_{MAX}}$ *roughly!!!*

2. "Orthogonal" Subcarriers and OFDM



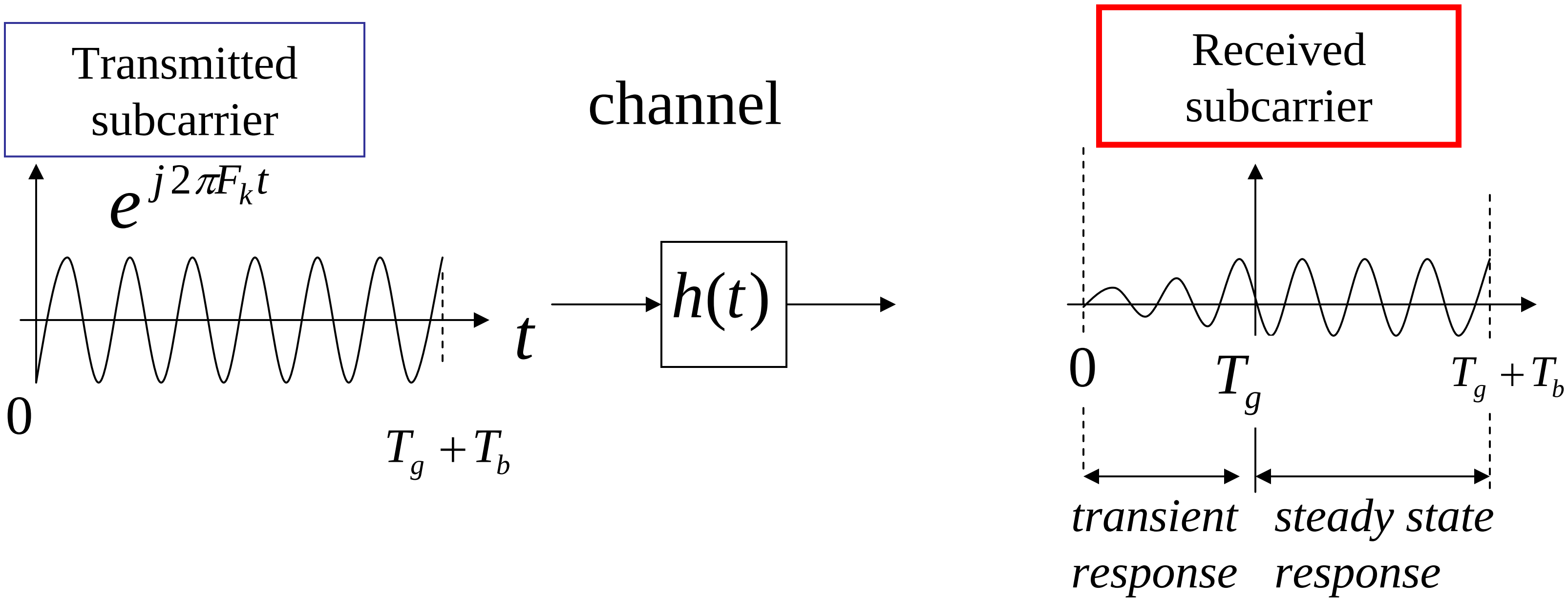


Choose: $\boxed{\Delta F = \dfrac{1}{T_b}}$

Orthogonality:

$$\frac{1}{T_b}\int_{t_0}^{t_0+T_b} e^{j2\pi F_k t} e^{-j2\pi F_\ell t} dt = \frac{1}{T_b}\int_{t_0}^{t_0+T_b} e^{j2\pi(k-\ell)\Delta F t} dt = \begin{cases} 1 & \text{if } k = \ell \\ 0 & \text{if } k \neq \ell \end{cases}$$

Since the channel is Linear and Time Invariant (at least for the duration of the frame), the exponentials $e^{j2\pi F_k t}$ are still orthogonal at the receiver <u>in steady state</u>, ie after the transient has died.



$$H(F_k)e^{j2\pi F_k t}$$

$$T_g \le t \le T_g + T_b$$

still orthogonal at the receiver!!!

Each OFDM symbol is generated in discrete time.

Let

- F_S be the sampling frequency;
- $N > N_F$ be the number of data samples in each symbol;
- $\Delta F = 1/\left(N\, T_S\right) = F_S / N$ the subcarriers spacing
- $A=1/N$

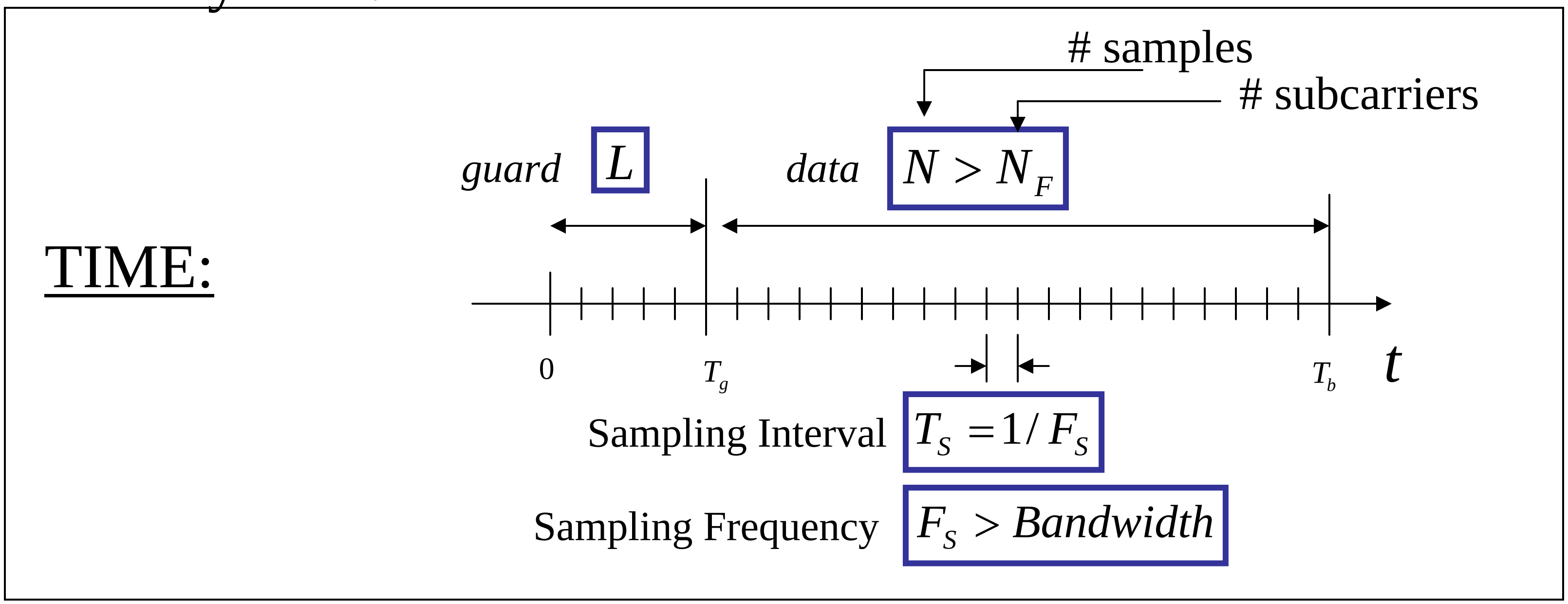
Then:

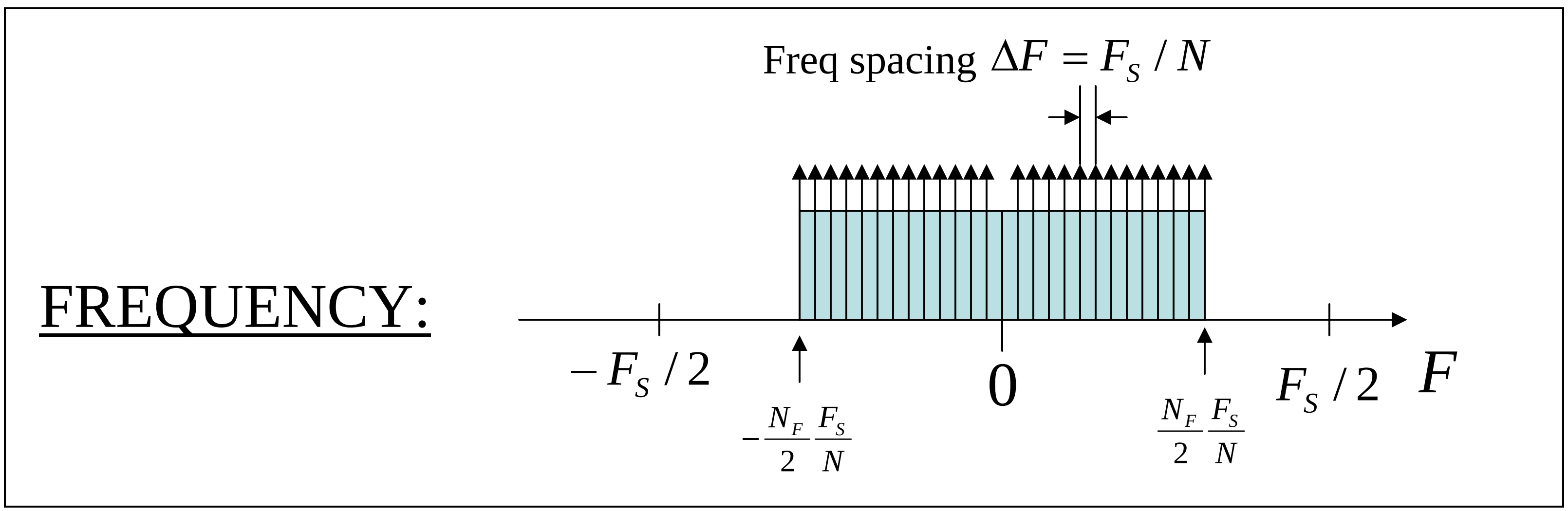
$$x(nT_S) = \frac{1}{N} \sum_{k=-\frac{N_F}{2}}^{\frac{N_F}{2}} c_k e^{j2\pi k \frac{\Delta F}{F_S}(n-L)} = \frac{1}{N} \sum_{k=-\frac{N_F}{2}}^{\frac{N_F}{2}} c_k e^{jk\frac{2\pi}{N}(n-L)} \quad n = 0,..,L+N-1$$

With $T_g = L \times T_S$ the guard time.

4. Generating the OFDM symbol using the IFFT

OFDM Symbol: discrete time





This can be written as

$$x[n+L] = \frac{1}{N}\sum_{k=-\frac{N_F}{2}}^{\frac{N_F}{2}} c_k e^{jk\frac{2\pi}{N}n}$$

$$= \frac{1}{N}\sum_{k=1}^{\frac{N_F}{2}} c_k e^{jk\frac{2\pi}{N}n} + \frac{1}{N}\sum_{k=-\frac{N_F}{2}}^{-1} c_k e^{j(N+k)\frac{2\pi}{N}n}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X[k] e^{jk\frac{2\pi}{N}n} = IFFT\{X[k]\}$$

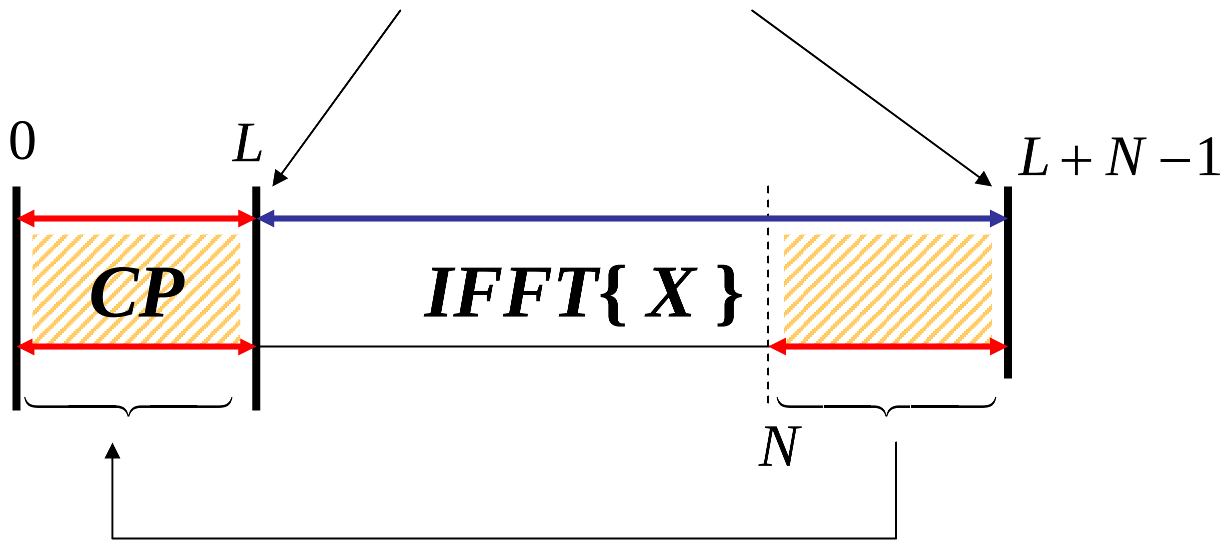
Where:

$X[k] = c_k,$	$k = 1,\ldots, N_F/2$	positive subcarriers
$X[N+k] = c_k,$	$k = -1,\ldots,-N_F/2$	negative subcarriers

$X[k] = 0,$ otherwise

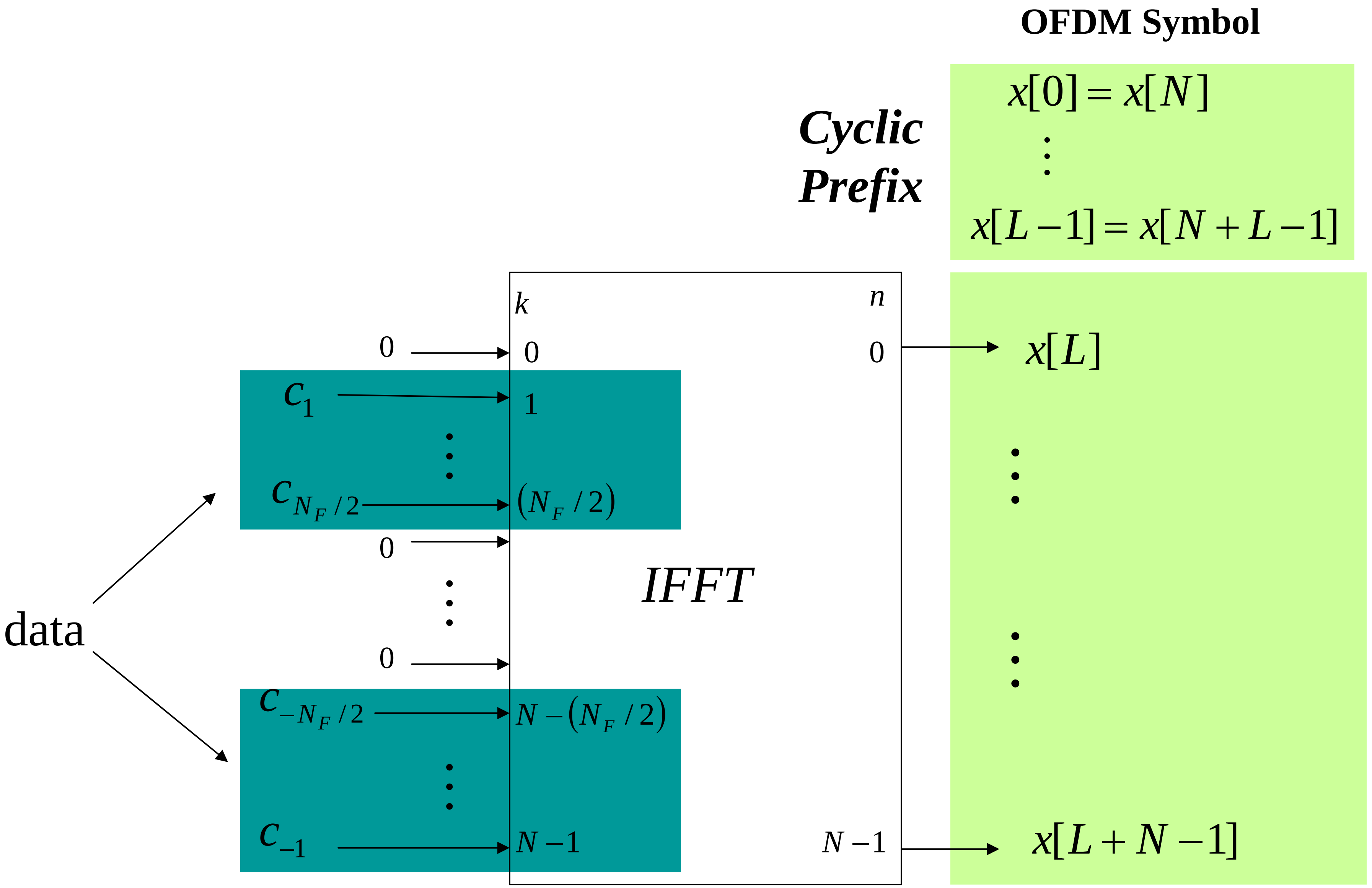
Guard Time with Cyclic Prefix (CP)

$$x[L],\ldots,x[L+N-1] = IFFT\{X[k], k=0,\ldots,N-1\}$$



CP from the periodicity:

$$x[n] = x[N+n] \quad \Rightarrow \quad \begin{aligned} &x[0] = x[N] \\ &x[1] = x[N+1] \\ &\ldots \\ &x[L-1] = x[L+N-1] \end{aligned}$$



OFDM Symbol
Cyclic Prefix
$x[0]=x[N]$
$x[L-1]=x[N+L-1]$
data
c_1
$c_{N_F/2}$
$c_{-N_F/2}$
c_{-1}
0
0
0
k
n
1
$(N_F/2)$
$N-(N_F/2)$
$N-1$
IFFT
$x[L]$
$x[L+N-1]$

Summary of OFDM

… and relevant parameters

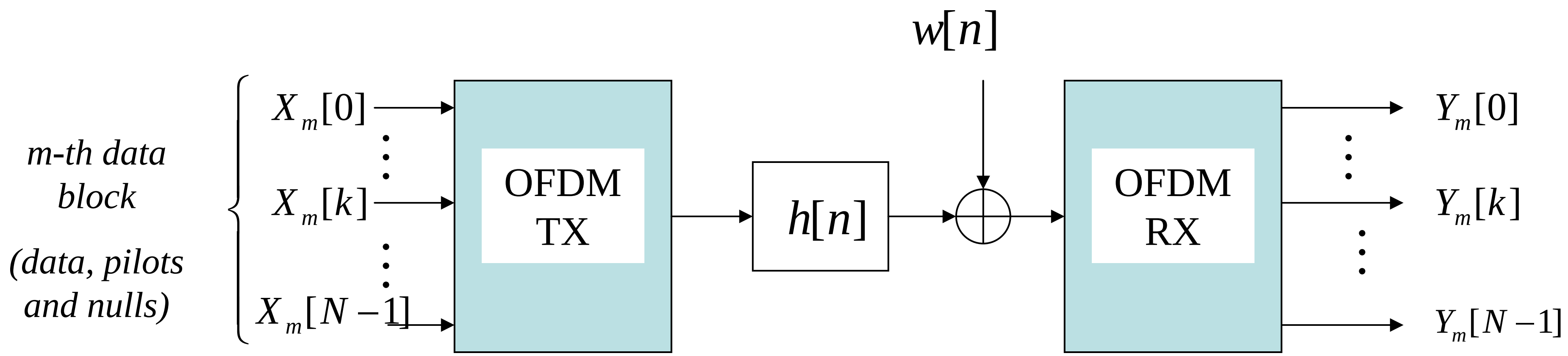
Channel (given parameters):

1.	Max Time Spread	**t_MAX** in sec
2.	Doppler Spread	**F_D** in Hz
3.	Bandwidth	**BW** in Hz

OFDM (design parameters):

1.	Sampling Frequency	**Fs**>BW in Hz
2.	Cyclic Prefix	**L** > t_MAX * Fs, integer
3.	FFT size (power of 2)	4*L<**N**<<Fs/F_D, integer
4.	Number of Carriers	**NF**=[N*BW/Fs], integer

Test in Simulation:



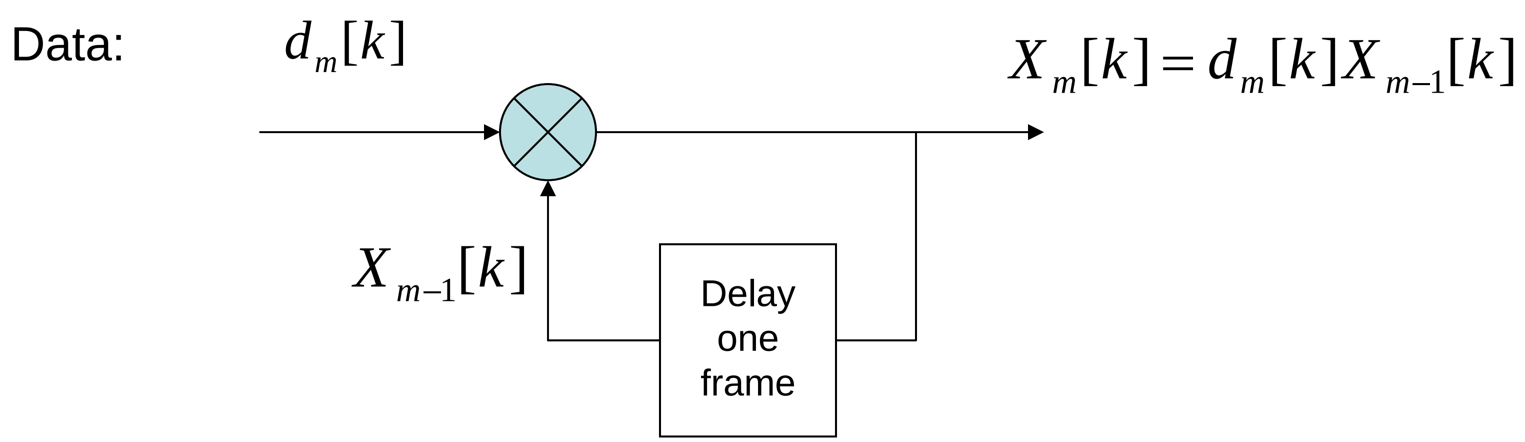
Recall that for every transmitted block of data we receive

$$Y_m[k] = H_m[k] X_m[k] + W_m[k]$$

With $H_m[k]$ the frequency response of the channel within the time block

In order to estimate $X_m[k]$ we need to know the channel.

A way to avoid it is to use differential encoding:



With QPSK it is equivalent to accumulating the phase

$$\angle X_m[k] = \angle X_{m-1}[k] + \angle d_m[k]$$

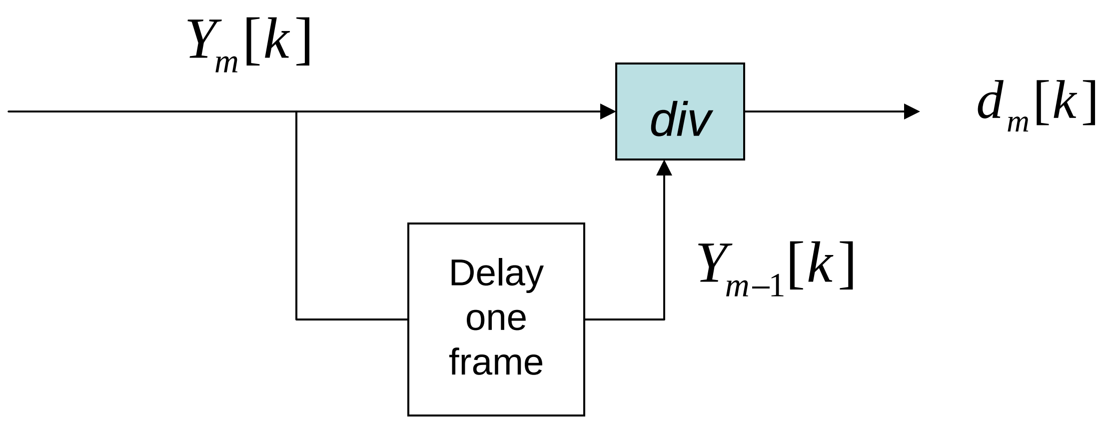
At the receiver

$$\hat{d}_m[k] = \frac{Y_m[k]}{Y_{m-1}[k]} = \frac{H_m[k]X_m[k] + W_m[k]}{H_{m-1}[k]X_{m-1}[k] + W_{m-1}[k]} \cong d_m[k]$$

Provided:

a. The channel changes slowly $H_m[k] \cong H_{m-1}[k] \neq 0$

b. High SNR



Lab 4: Single Carrier vs. OFDM Modulation

Goal: In this Lab we want to compare Single Carrier (SC) modulation with Orthogonal Frequency Division Multiplexing (OFDM), with Differential Modulation .

1. Using the Simulink model **test_SC.mdl** see the received signal for various values of the symbol rate

$$F_S = 2.0,\ 20.0,\ 200.0,\ 2000.0\ kHz$$

As the symbol rate increases, notice the effect of Inter Symbol Interference in the scattering plot.

2. Repeat the same with the Simulink model **test_OFDM.mdl and** see the received signal for the same values of the symbol rate. Notice how you can increase the data rate and still be able to demodulate the received signal.

Example: IEEE 802.11a (WiFi)

1. Parameters
2. Simulink Example
3. “Frame based” and “Sample based” signals

Parameters of IEEE802.11a:

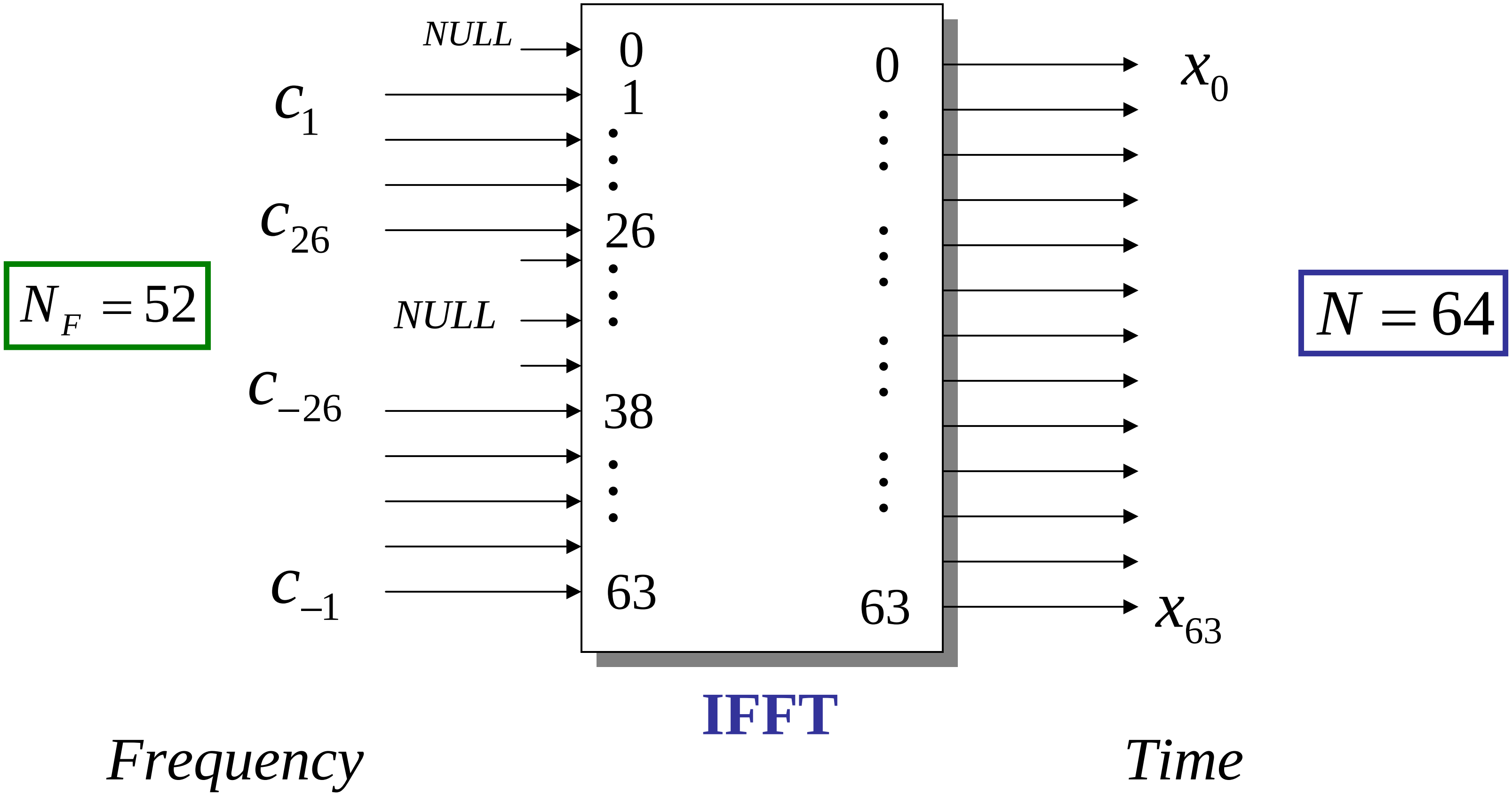
<u>**Channel**</u> (given parameters):

1. Max Time Spread **t_MAX=0.5 microsec**
2. Doppler Spread **F_D=50Hz**
3. Bandwidth **BW=16MHz**

OFDM (design parameters):

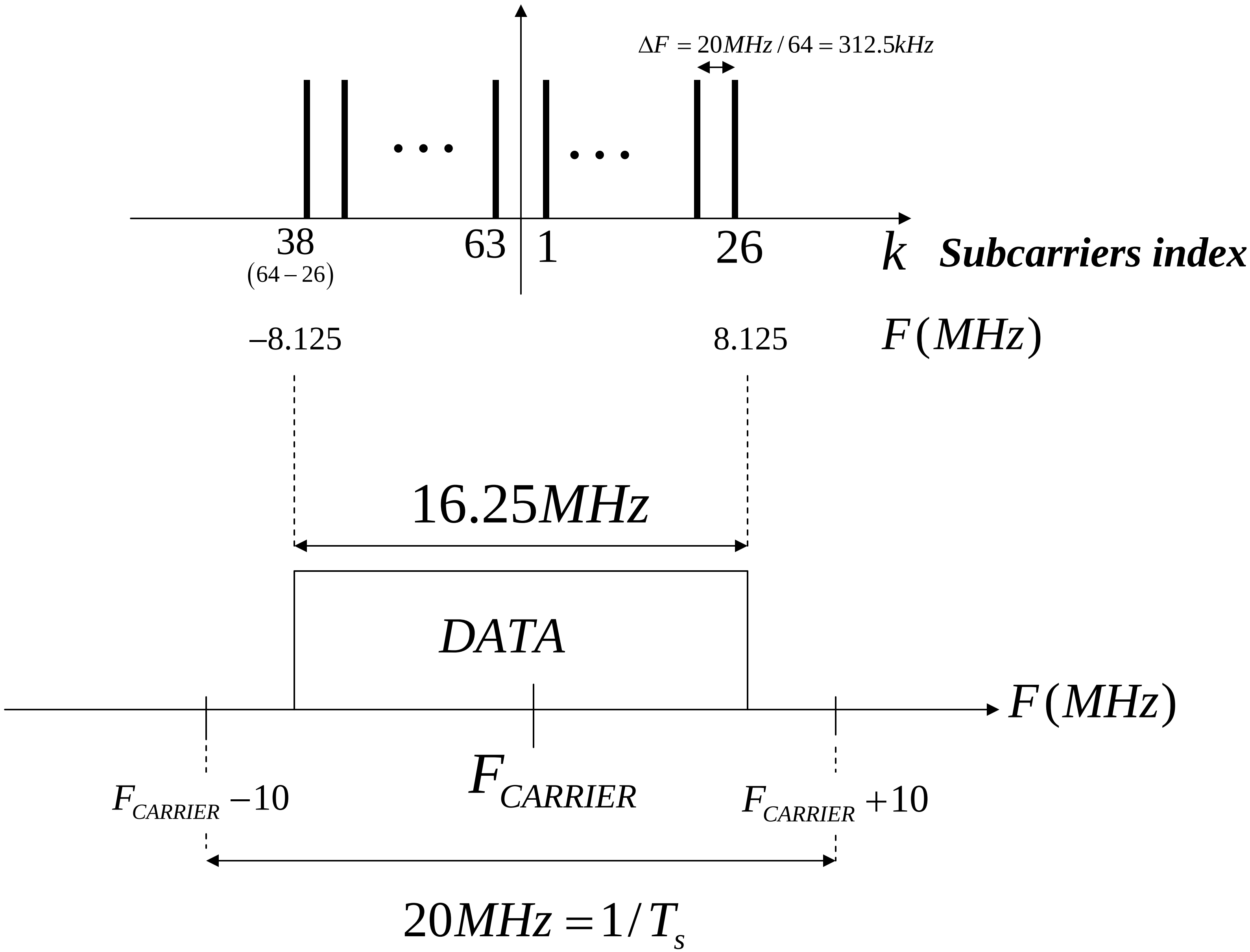
1. Sampling Frequency **Fs=20MHz**
2. Cyclic Prefix **L=16** > 0.5*20=10
3. FFT size (power of 2) **N=64**<<20e06/50
4. Number of Carriers **NF=52=[64*16/20]**

Sub-carriers: (48 data + 4 pilots) + (12 nulls) = 64

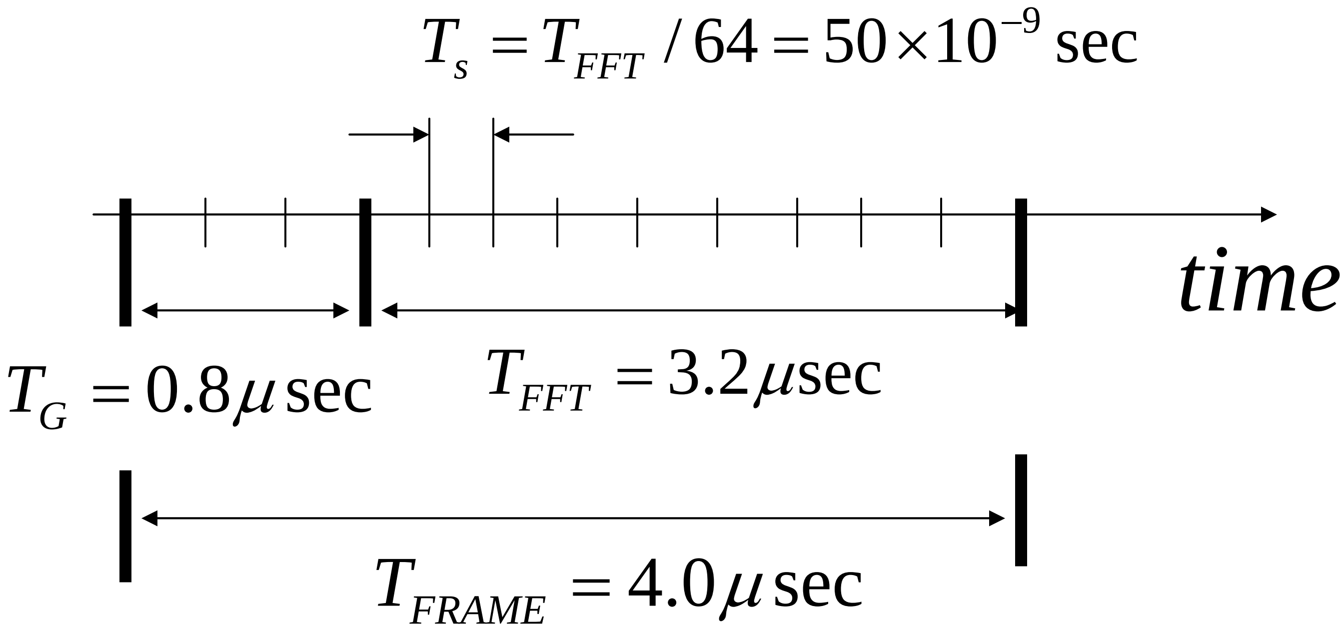


Pilots at: -21, -7, 7, 21

Frequencies:



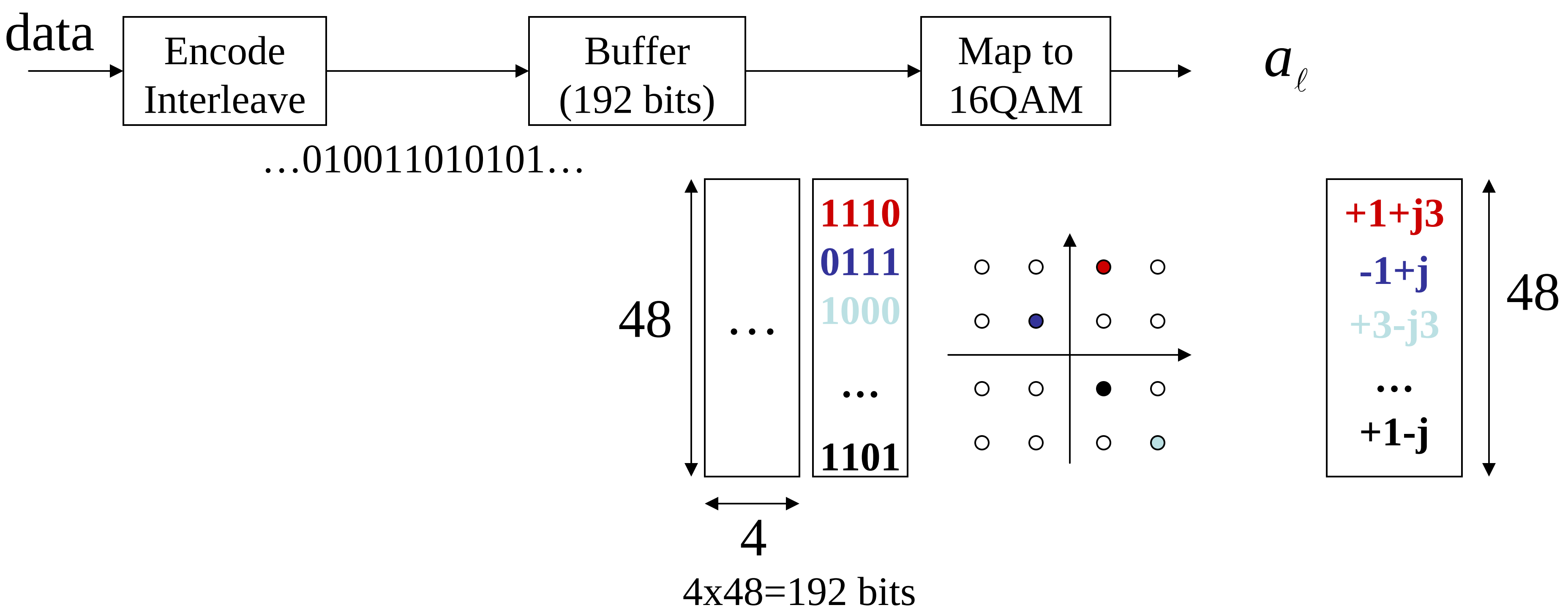
Time Block:



$T_s = T_{FFT} / 64 = 50 \times 10^{-9}$ sec
time
$T_G = 0.8 \mu$sec
$T_{FFT} = 3.2 \mu$sec
$T_{FRAME} = 4.0 \mu$sec

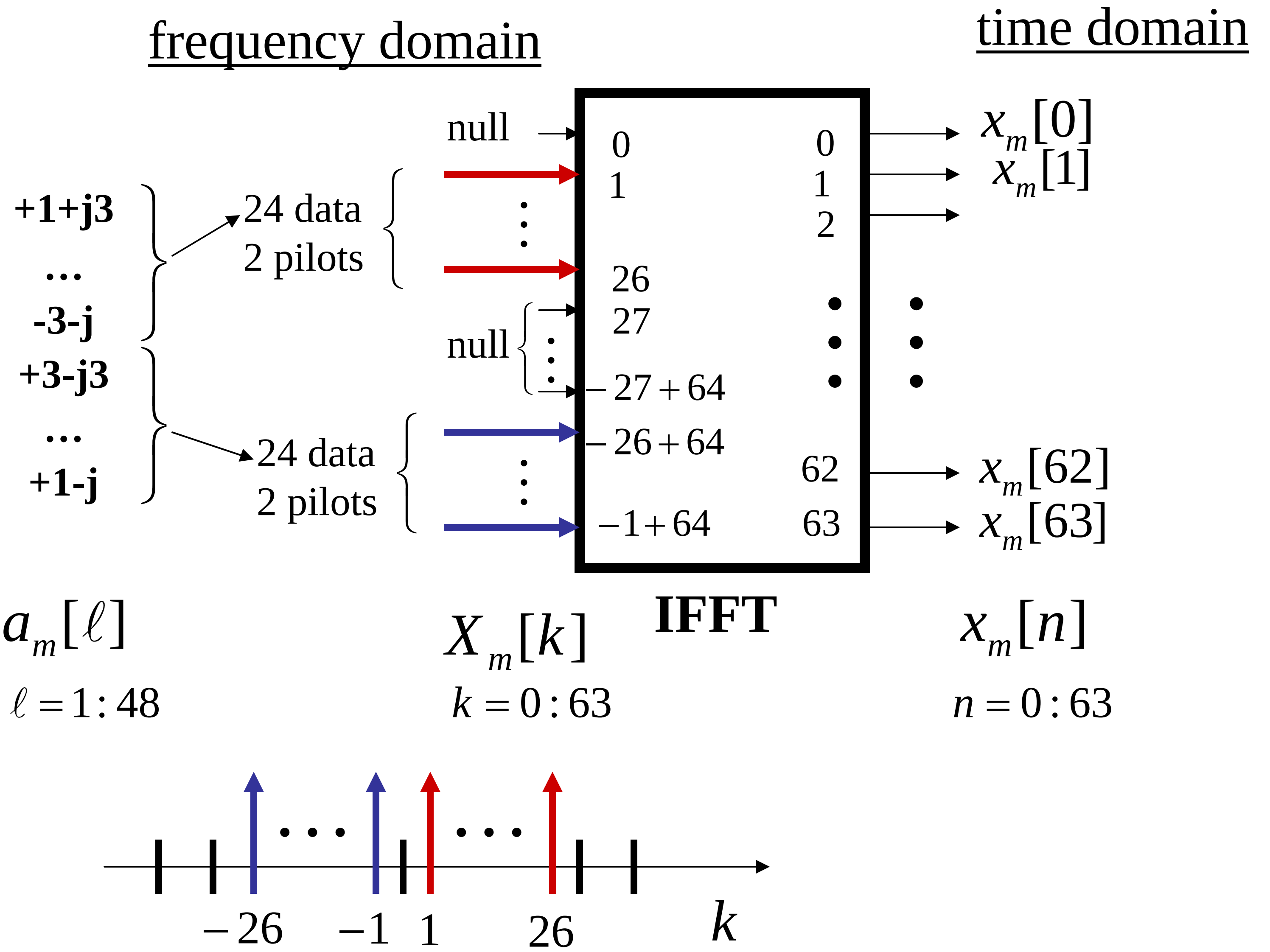
Overall Implementation (IEEE 802.11a with 16QAM).

1. Map encoded data into blocks of **192** bits and **48** symbols:



Overall Implementation (IEEE 802.11a with 16QAM).

2. Map each block of **48** symbols into **64** samples



Simulink Example

To make it simpler:

1. let the number of carrier be the same as the FFT length, ie **N=NF;**

2. Use Differential QPSK Encoding and Decoding, so that we do not need to estimate the channel's frequency response.

Initial Callback function:

% OFDM parameters (IEEE802.11a)

```
Fs=20e6;   % symbol data rate (uncoded) in Hz
N=64;      % FFT sample size
L=16;      % Cyclic Prefix sample length
```

% Channel Parameters

% 1. Doppler Spread

```
FC=5.0;    % carrier freq. in GHz
v=50;      % speed in km/h
FD=v*FC;   % doppler freq in Hz
```

%2. Time Spread

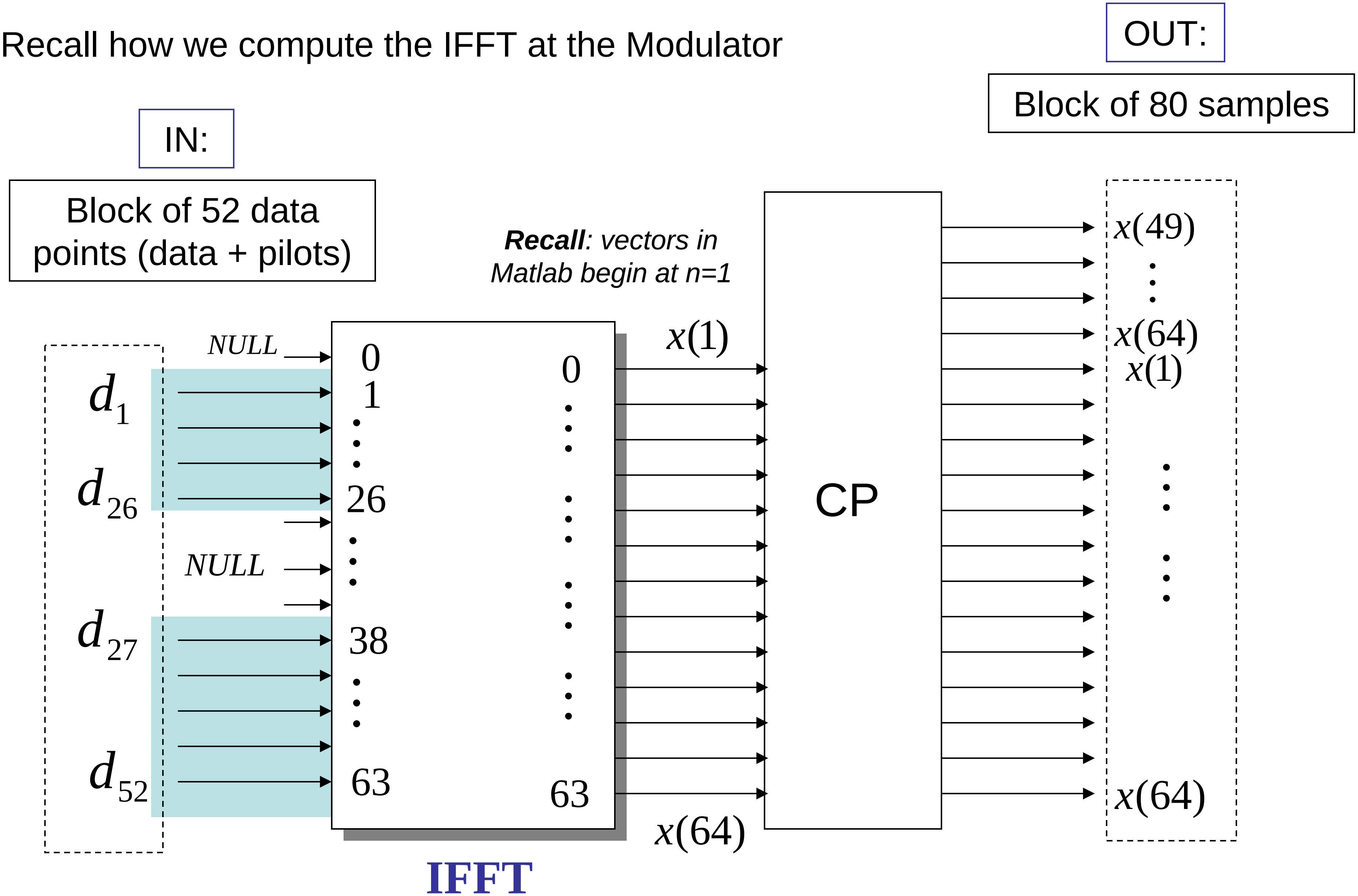
```
tau=[0, 0.1, 0.4]*1e-6;   % time delays in seconds
P=[0, -2, -4];            % attenuations in dB
```

% Additive Noise

```
SNRdB=20;   % dB
```

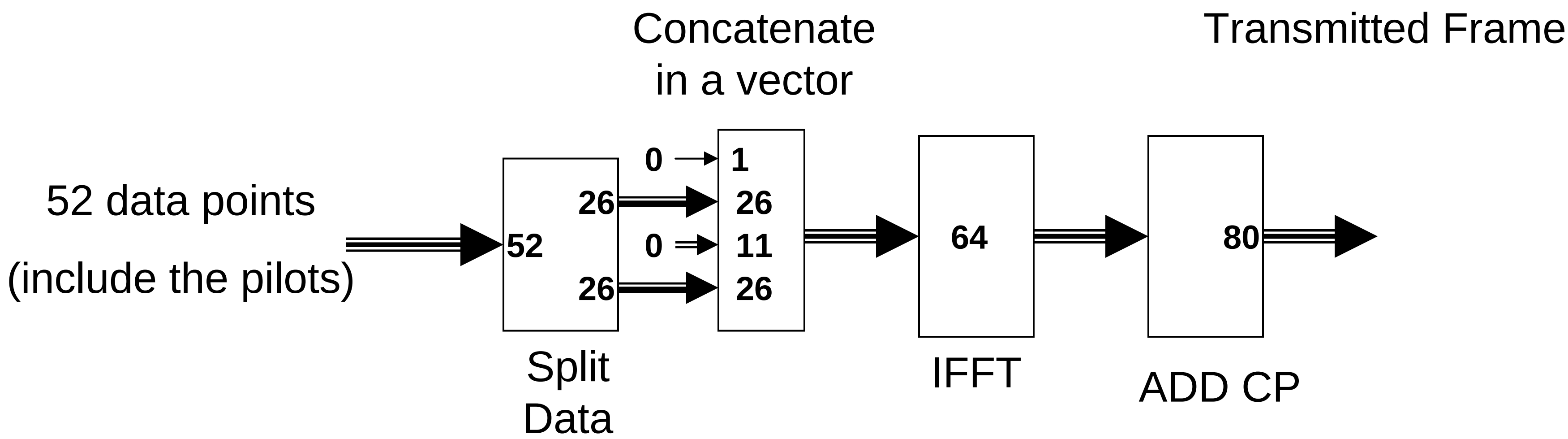
Simulink Implementation of OFDM IEEE802.11a

Recall how we compute the IFFT at the Modulator

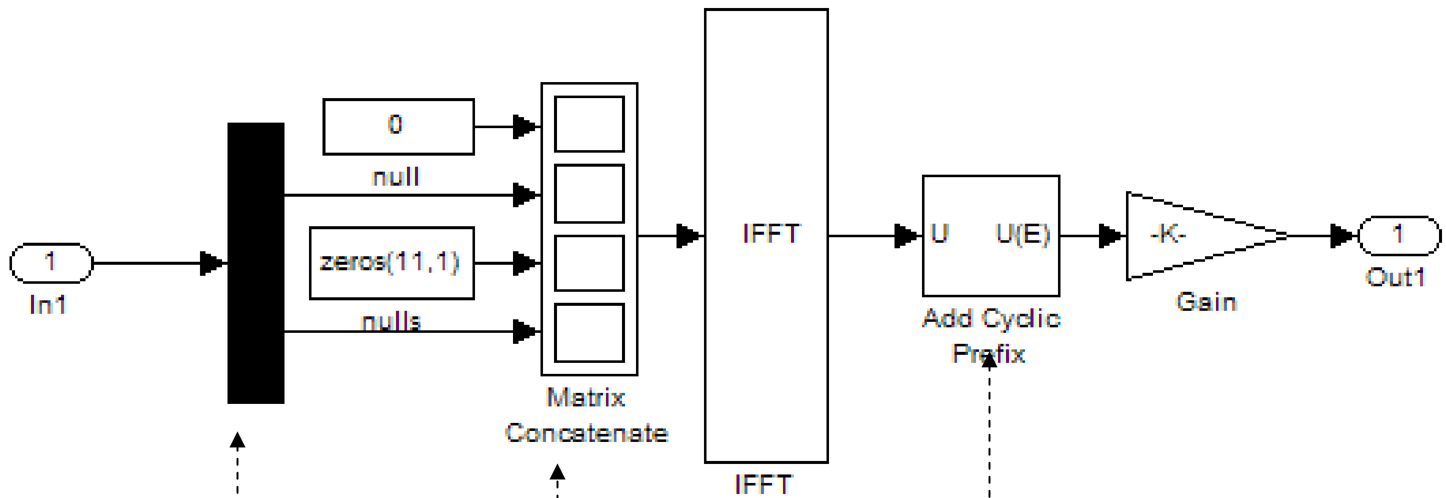


Vector operations.

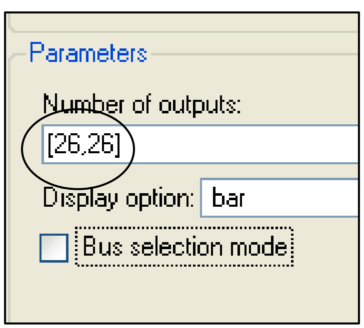
OFDM Modulator:



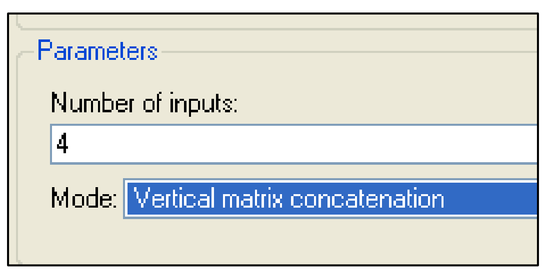
OFDM Modulator:



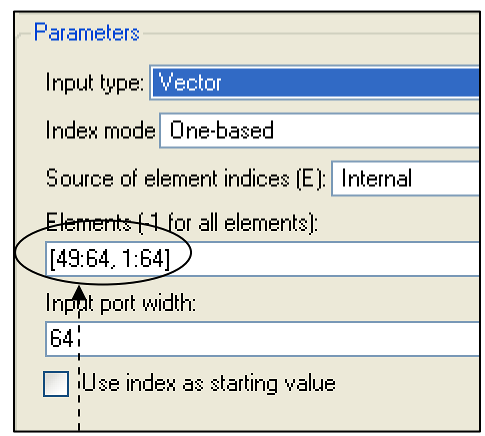
Simulink > Commonly Used Blocks > demux



Simulink > Math Operations > Matrix Concatenate



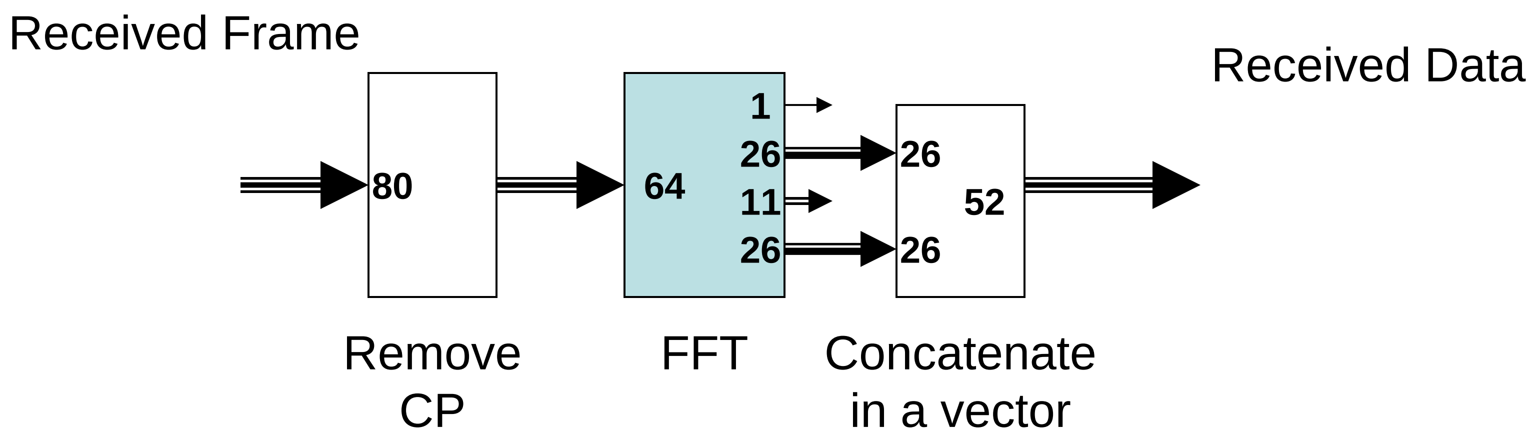
Simulink > Signal Routing > Selector

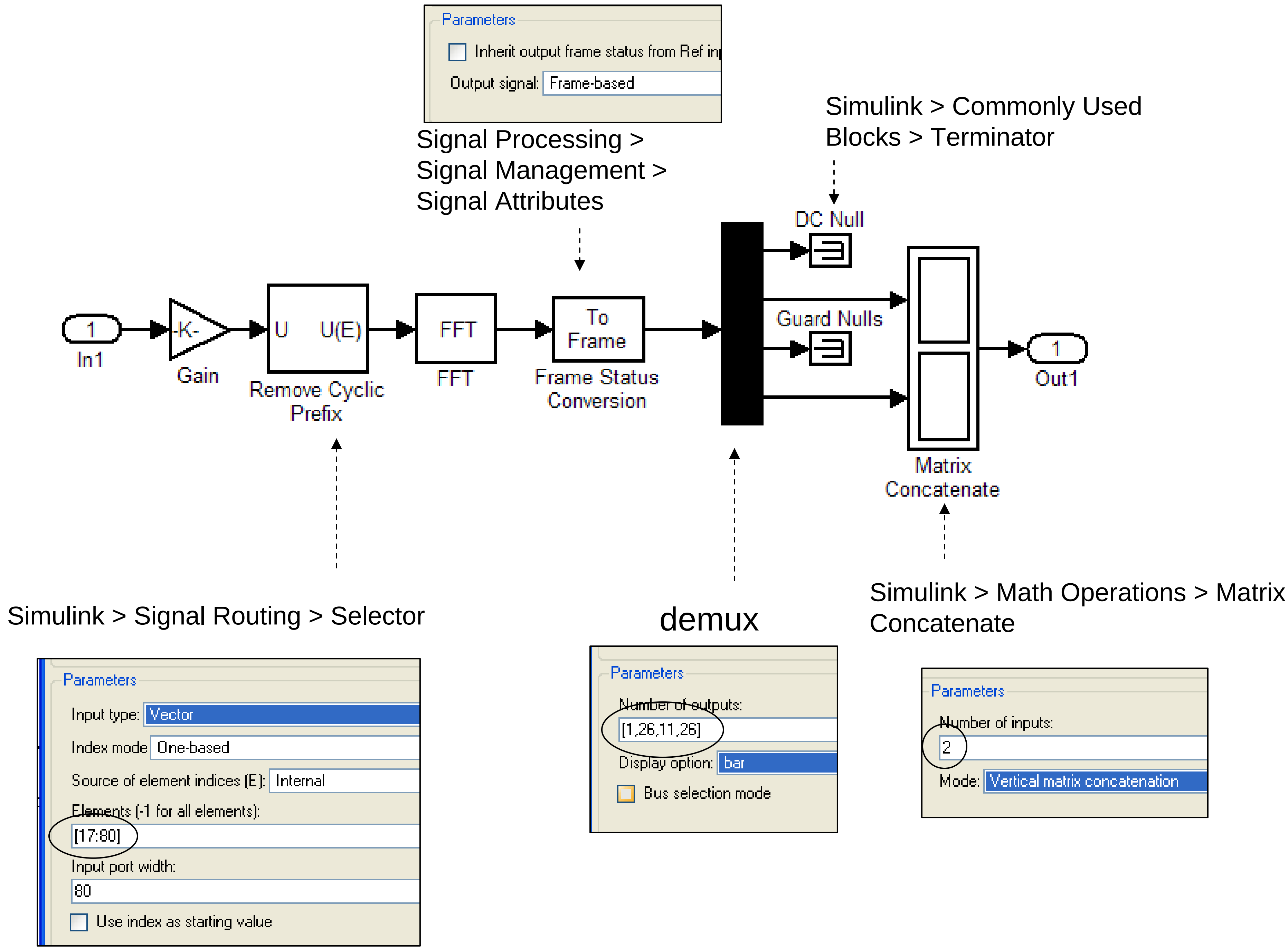


repeat last 16 entries

[49:64, 1:64]

OFDM Demodulator:



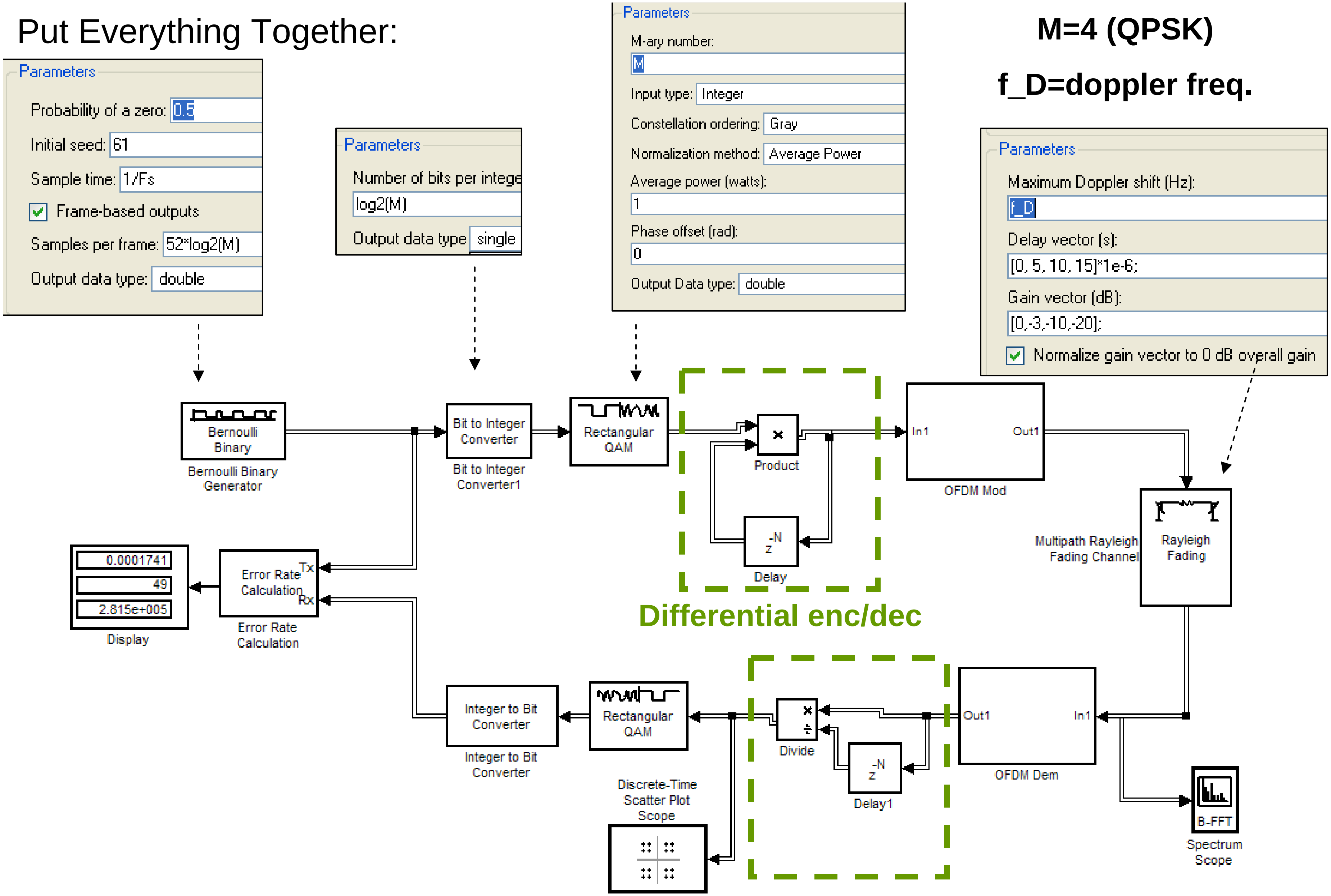


Parameters
Inherit output frame status from Ref in
Output signal: Frame-based
Signal Processing > Signal Management > Signal Attributes
Simulink > Commonly Used Blocks > Terminator
DC Null
Guard Nulls
1
In1
-K-
Gain
U U(E)
Remove Cyclic Prefix
FFT
FFT
To Frame
Frame Status Conversion
Matrix Concatenate
1
Out1
Simulink > Signal Routing > Selector
demux
Simulink > Math Operations > Matrix Concatenate
Parameters
Input type: Vector
Index mode One-based
Source of element indices (E): Internal
Elements (-1 for all elements):
[17:80]
Input port width:
80
Use index as starting value
Parameters
Number of outputs:
[1,26,11,26]
Display option: bar
Bus selection mode
Parameters
Number of inputs:
2
Mode: Vertical matrix concatenation

Put Everything Together:

M=4 (QPSK)

f_D=doppler freq.



3. “Frame based” and “Sample based”

In Simulink a signal can be

- Frame based
- Sample based

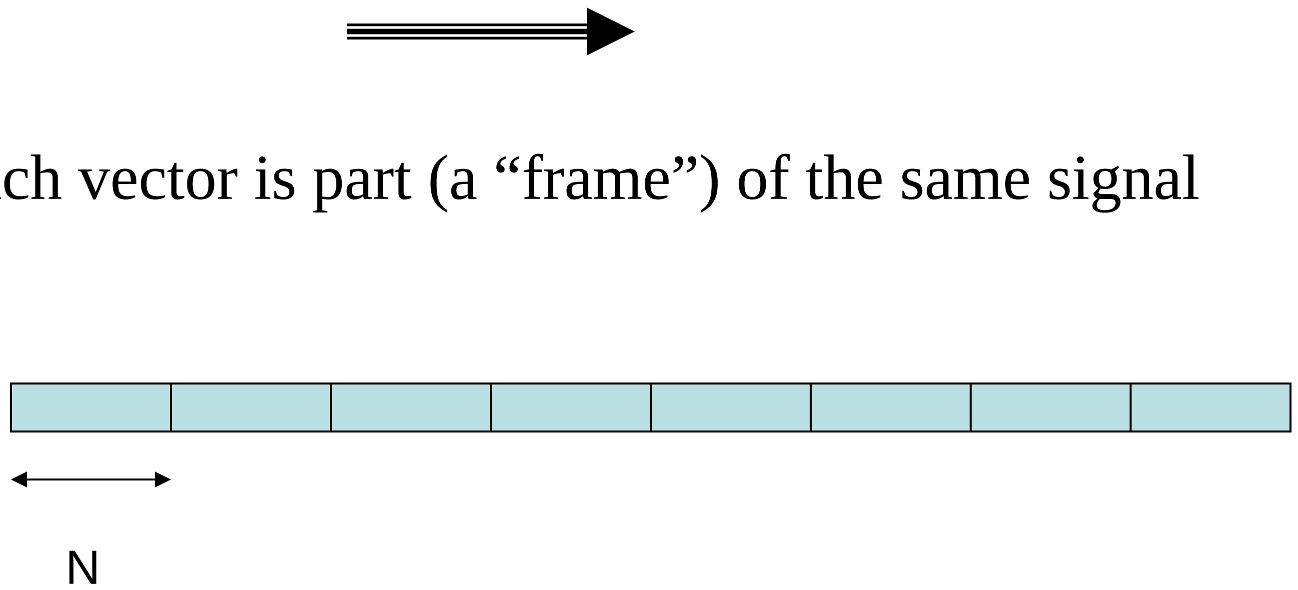
This is particularly important when the signals are vectors.

- Some blocks want the input to be Frame based, others want it Sample based, others don’t care
- When you have to process a large number of data, it is more efficient to process blocks of samples, using Frames, rather than one sample at a time.

“Frame based”

Frame Based [Nx1]

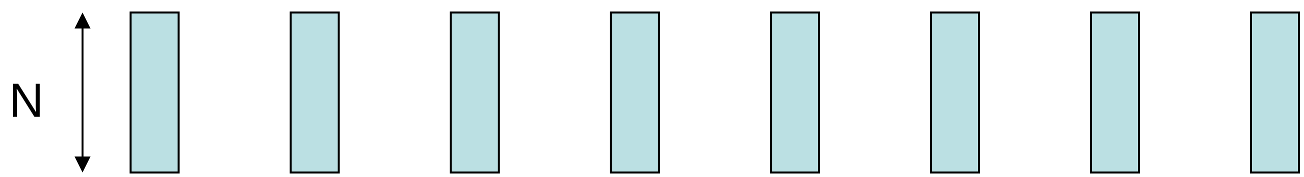
Each vector is part (a “frame”) of the same signal



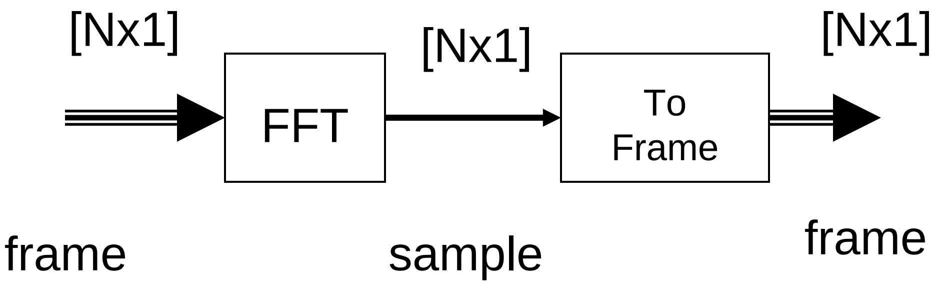
“Sample based”

Sample Based [Nx1]

There are N different signals. Each element of the vector belongs to a different signal.



Typical Example: the FFT block



IEEE 802.16 Standard

IEEE 802.16 2004:

Part 16: Air Interface for Fixed Broadband Wireless Access Systems

From the Abstract:

• It specifies air interface for **fixed** Broadband Wireless Access (BWA) systems supporting multimedia services;

• MAC supports point to multipoint with optional mesh topology;

• multiple physical layer (PHY) each suited to a particular operational environment:

IEEE 802.16-2004 Standard

Table 1 (Section 1.3.4) **Air Interface Nomenclature**:

- WirelessMAN-SC, Single Carrier (SC), Line of Sight (LOS), 10-66GHz, TDD/FDD
- WirelessMAN-SCa, SC, 2-11GHz licensed bands,TDD/FDD
- WirelessMAN OFDM, 2-11GHZ licensed bands,TDD/FDD
- WirelessMAN-OFDMA, 2-11GHz licensed bands,TDD/FDD
- WirelessHUMAN 2-11GHz, unlicensed,TDD

MAN: Metropolitan Area Network

HUMAN: High Speed Unlicensed MAN

IEEE 802.16e 2005:

Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems

Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands

and

Corrigendum 1

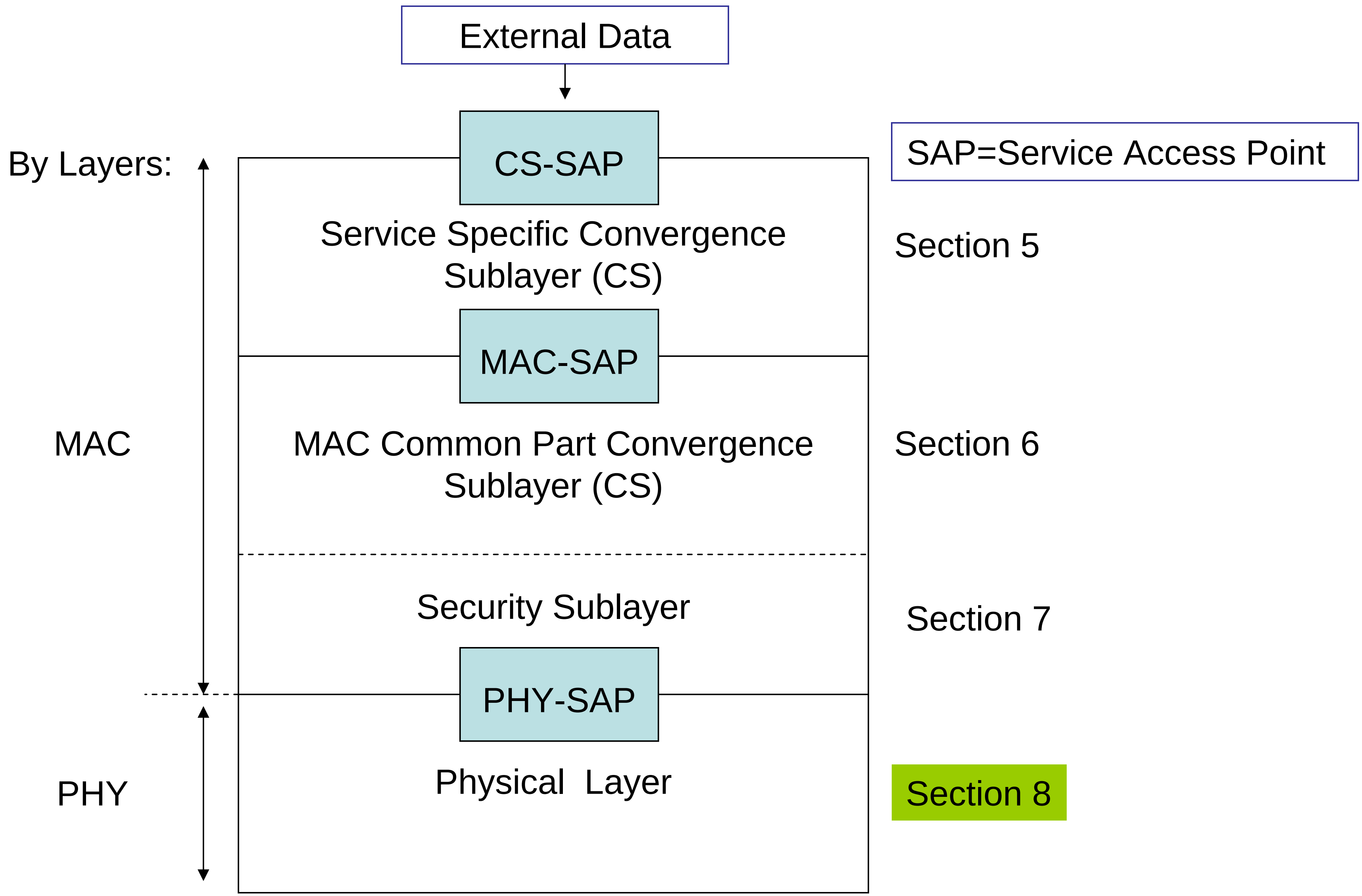
Scope (Section 1.1):

• it enhances IEEE 802.16-2004 to support mobility at vehicular speed, for combined fixed and mobile Broadband Wireless Access;

• higher level handover between base stations;

• licensed bands below 6GHz.

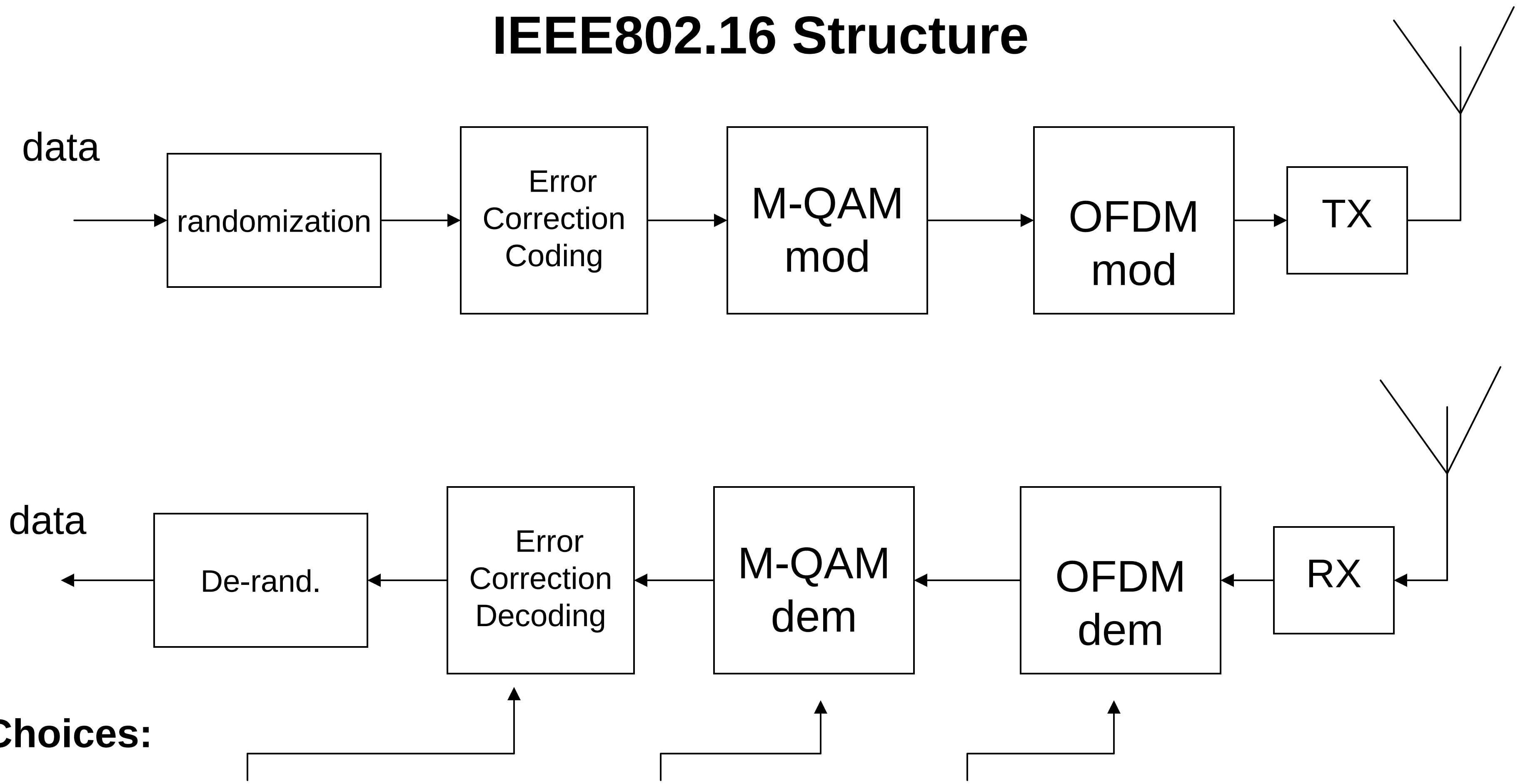
IEEE 802.16-2004: Reference Model (Section 1.4), Figure 1



Parameters for IEEE 802.16 (OFDM only)

	802.16-2004	**802.16e-2005**
Frequency Band	2GHz-11GHz	2GHz-11GHz fixed 2GHz-6GHz mobile
OFDM carriers	OFDM: 256 OFDMA: 2048	OFDM: 256 OFDMA: 128, 256, 512,1024, 2048
Modulation	QPSK, 16QAM, 64QAM	QPSK, 16QAM, 64QAM
Transmission Rate	1Mbps-75Mbps	1Mbps-75Mbps
Duplexing	TDD or FDD	TDD or FDD
Channel Bandwidth	(1,2,4,8)x1.75MHz (1,4,8,12)x1.25MHz 8.75MHz	(1,2,4,8)x1.75MHz (1,4,8,12)x1.25MHz 8.75MHz

IEEE802.16 Structure



Choices:

Coding rates
1/2
2/3
3/4
5/6

M-QAM
2
4
16
64

OFDM carriers
256
512
1024
2048

Channel B/width
1.25 MHz
5 MHz
10 MHz
…

OFDM and OFDMA (Orthogonal Frequency Division Multiple Access)

• Mobile WiMax is based on OFDMA;

• OFDMA allows for subchannellization of data in both uplink and downlink;

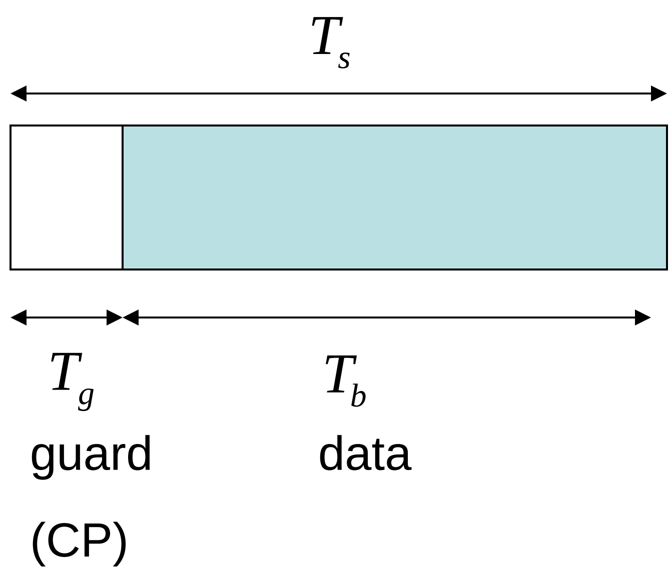
• Subchannels are just subsets of the OFDM carriers: they can use contiguous or randomly allocated frequencies;

• FUSC: Full Use of Subcarriers. Each subchannel has up to 48 subcarriers evenly distributed through the entire band;

• PUSC: Partial Use of Subcarriers. Each subchannel has subcarriers randomly allocated within clusters (14 subcarriers per cluster) .

Section 8.3.2: OFDM Symbol Parameters and Transmitted Signal

OFDM Symbol



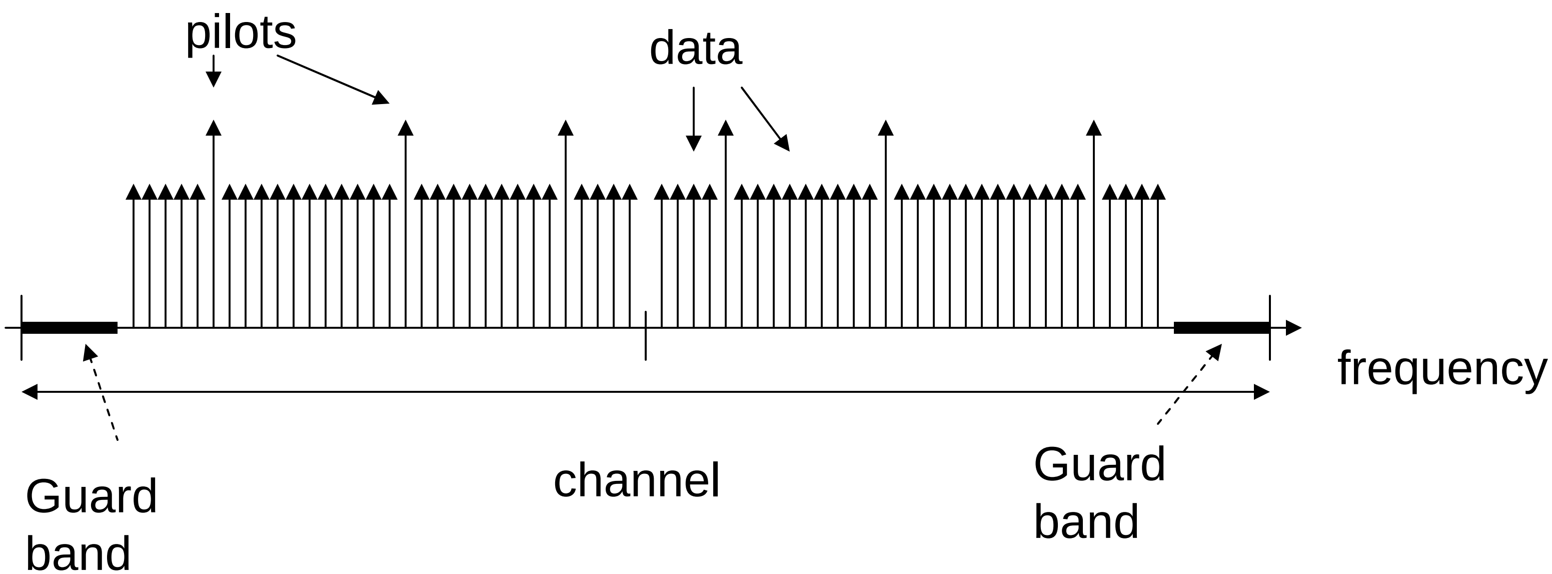
$$\boxed{\frac{T_g}{T_b} = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}}$$

Cyclic Prefix has a variable length. It represents a loss of

$$L_{CP} = 10 \log\left(T_b / (T_b + T_g)\right)$$

An OFDM Symbol is made of

- Data Carriers: data
- Pilot Carriers: synchronization and estimation
- Null Carriers: guard frequency bands and DC (at the modulating carrier)



$$N_{guards} = \text{ to provide frequency guards between channels}$$

$$N_{nulls} = N_{guards} + 1 \text{ (DC subcarrier is always zero)}$$

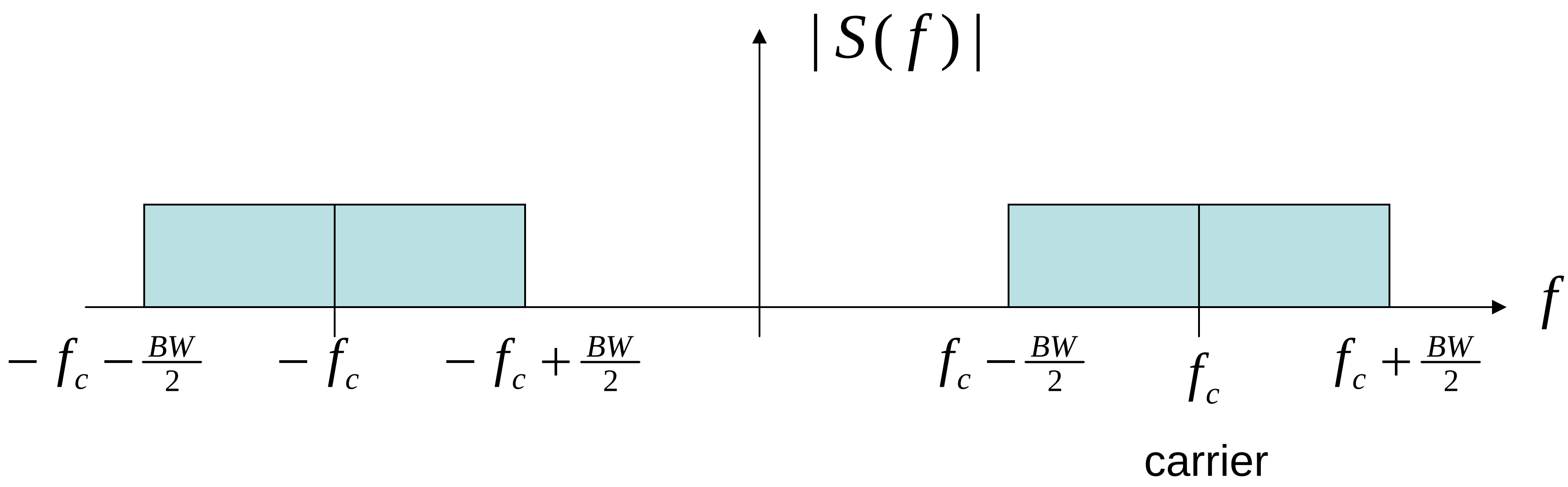
$$N_{pilots} = \text{ pilots for channel tracking and synchronization}$$

$$N_{data} = \text{ data subcarriers}$$

$$N_{used} = N_{pilots} + N_{data}$$

Recall the OFDM symbol is of the form

$$s(t) = \mathrm{Re}\left\{ e^{j2\pi f_c t} \sum_{\substack{k=-\frac{N_{used}}{2} \\ k\neq 0}}^{k=\frac{N_{used}}{2}} c_k e^{j2\pi k\Delta f\,(t-T_g)} \right\}$$



OFDM Subcarrier Parameters:

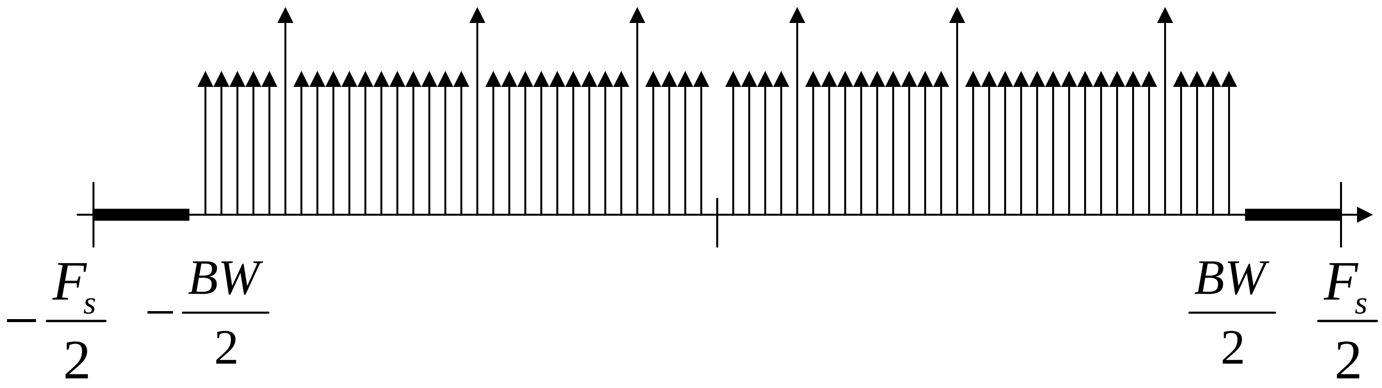
FFT size	256	128	512	1024	2048
N_used	200	108	426	850	1702
N_nulls	56	20	86	174	346
N_pilots	8	12	42	82	166
N_data	192	96	384	768	1536

Fixed WiMax

Fixed and Mobile WiMax

In 802.16 2004 we have

$$N_{FFT} = 256$$
$$N_{used} = 200$$



Frequency Spacing $\Delta f = 1/\text{data\_symbol\_length} = 1/T_b$

Sampling frequency $F_s = N_{FFT}\Delta f$

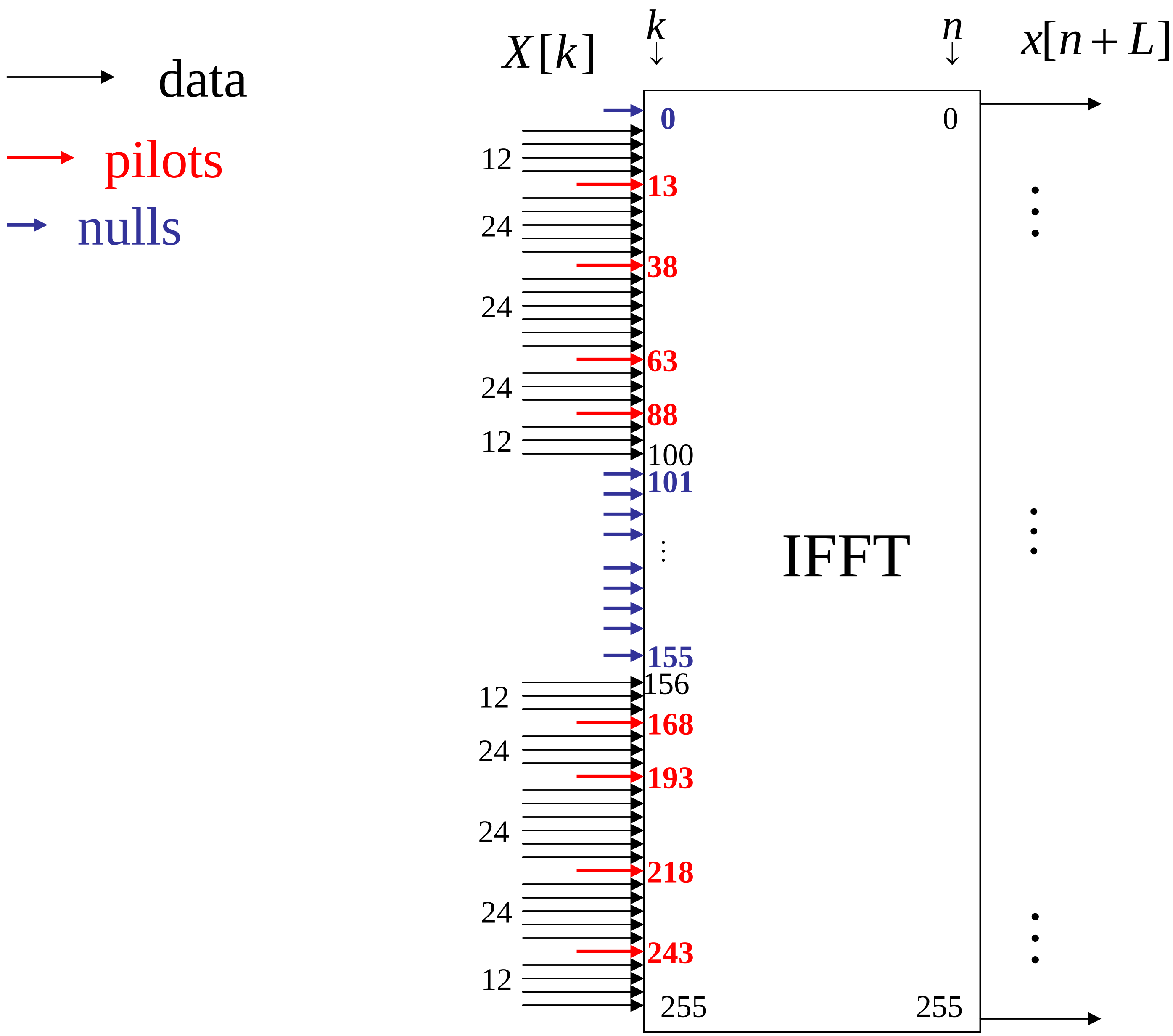
Bandwidth $BW = N_{used}\Delta f = \dfrac{N_{used}}{N_{FFT}} F_s$

Since we want the sampling rate to be integer multiple of a fixed rate (say 8kHz), the formula for the sampling rate is

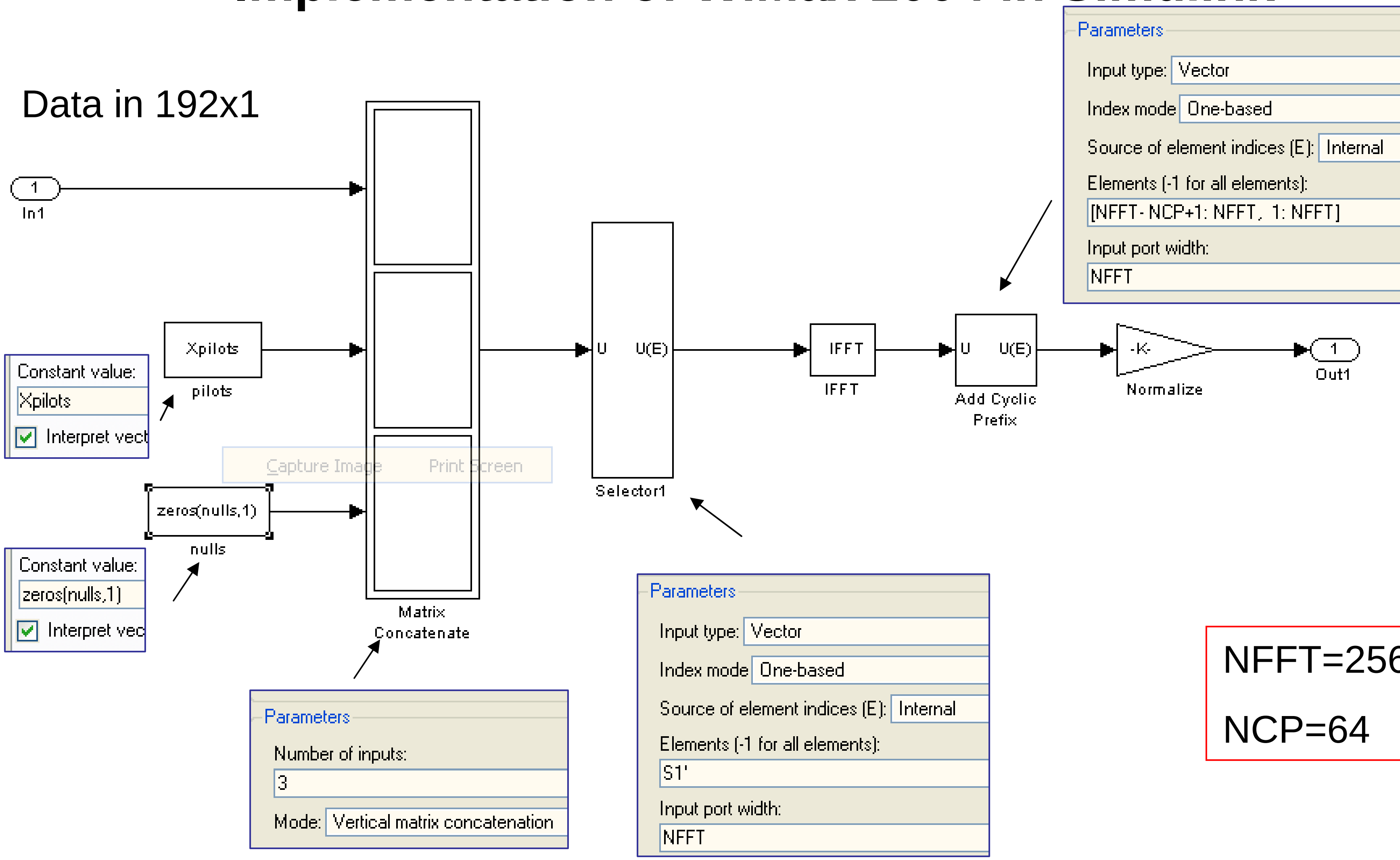
$$F_s = floor(n \times BW / 8{,}000) \times 8{,}000$$

With *n* a factor (Table 213) of the order $n = \frac{8}{7}, \frac{86}{75}, \frac{144}{125}, \frac{316}{275}, \frac{57}{50}, \approx \frac{256}{200}$

IEEE 802.16, with N=256



Implementation of WiMax 2004 in Simulink



NFFT=256

NCP=64

5. Error Correcting Coding

1. *Channel Capacity and Error Correction Coding*
2. *Block Coding*
3. *Code Shortening and Puncturing*
4. *Simulink Implementation of Block Codes*
5. *Convolutional Codes*
6. *Simulink Implementation of Convolutional Codes*
7. *Concatenated Codes in IEEE802.16*

1. *Channel Capacity and Error Correction Coding*

1.1 ***Channel Capacity***

1.2 ***Error Correction Coding***

1.3 ***Minimum SNR Requirement and Shannon Limit***

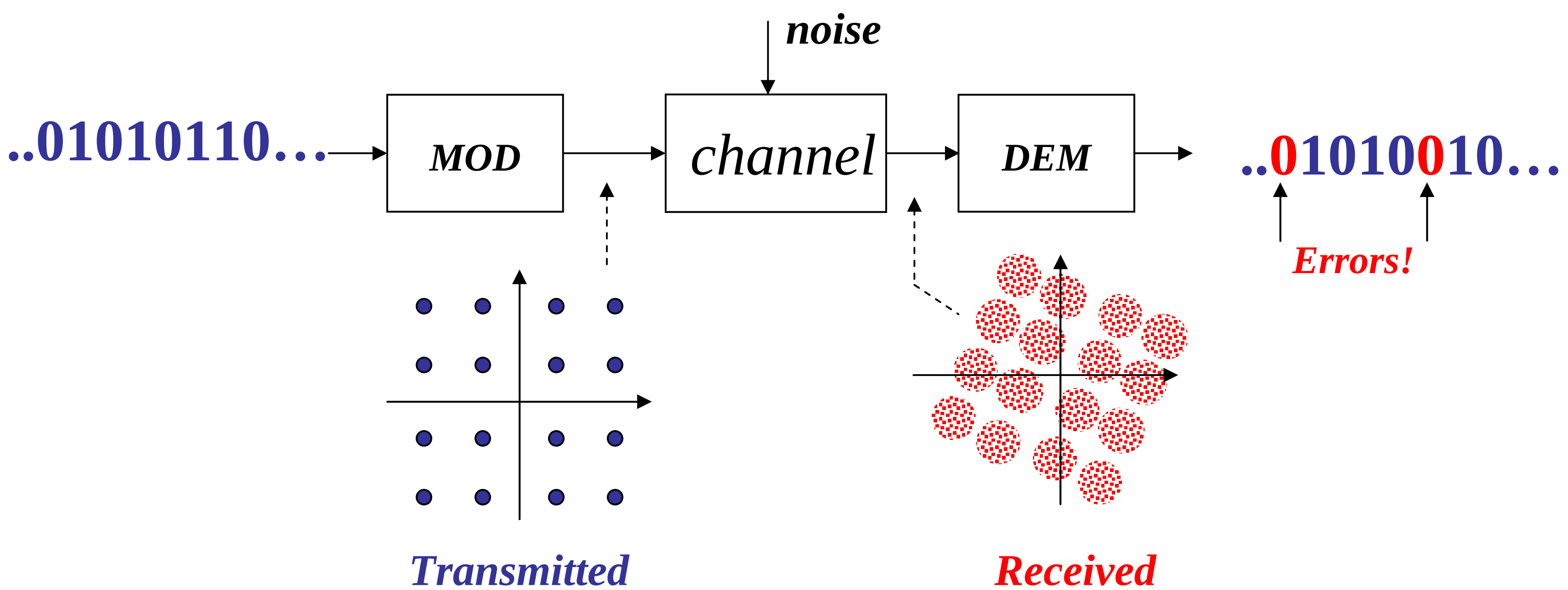
1. Channel Capacity

Problem: given a AWGN Channel, defined in terms of

- **Signal to Noise Ratio (*SNR*)**
- **Bandwidth *B***

and a desired Bit Error Rate (BER),

what is the **maximum data rate** we can transmit?



Define the **Channel Capacity**:

$$C = B\log_2(1+SNR) \quad \text{bits/sec}$$

Significance: if the transmitted data rate ***R<C*** then we can always find a coding scheme to make the bit error rate arbitrarily small.

Example:

SNRdB=20dB

B=10MHz

$$\Longrightarrow \quad C = 10^7 \log_2\left(1+10^{20/10}\right) = 66.5 Mbits/\sec$$

Problem: How do we encode the data to obtain an acceptable error rate?

Use Error Correction Coding!

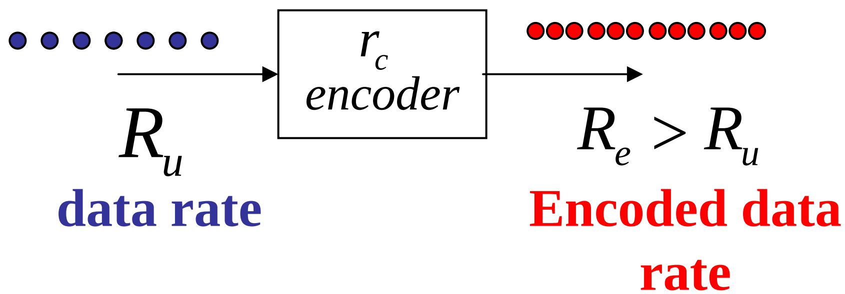
2. Error Correction Coding

<u>Purpose</u>: detect and Correct Errors by adding redundant information.

<u>How to Define an Encoder:</u>

- **code rate**

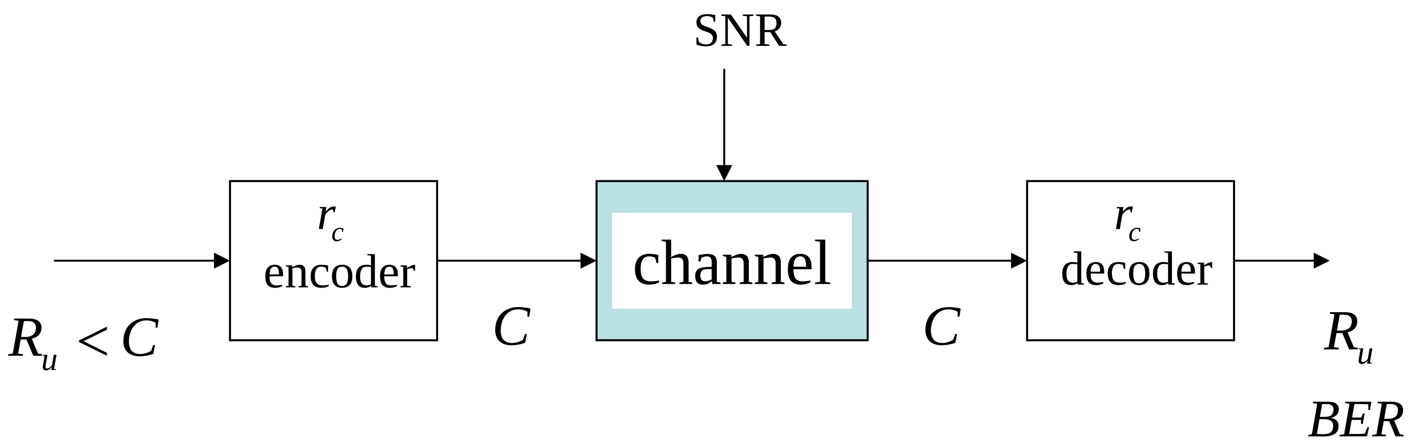
$$\boxed{r_c = \frac{R_u}{R_e} < 1}$$



- **number of errors corrected**

$$\boxed{BER = \frac{\#\,\text{bits in error}}{\text{total}\,\#\,\text{of bits}}}$$

Relation to Channel Capacity



Given

- a data rate $R_u < C$

- a desired Bit Error Rate (*BER*) arbitrarily small

<u>you can always find a code</u> with coding rate

$$r_c \leq \frac{R_u}{C} < 1$$

with desired *BER*.

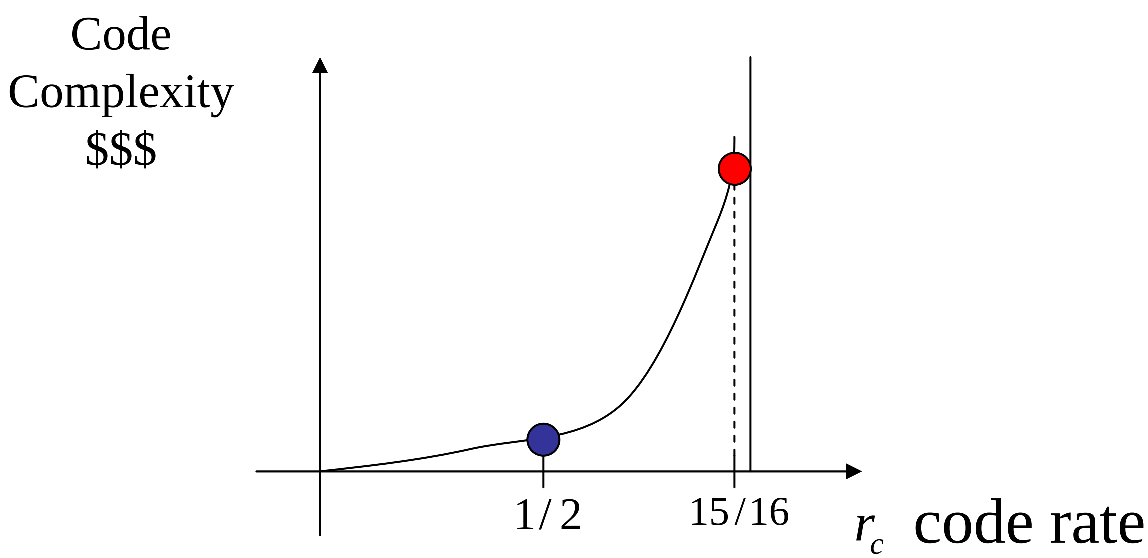
Clearly the Complexity goes up with the coding rate.

Example: see two situations with the same $BER = 10^{-5}$ and

$E_b / N_0 \approx 4dB$

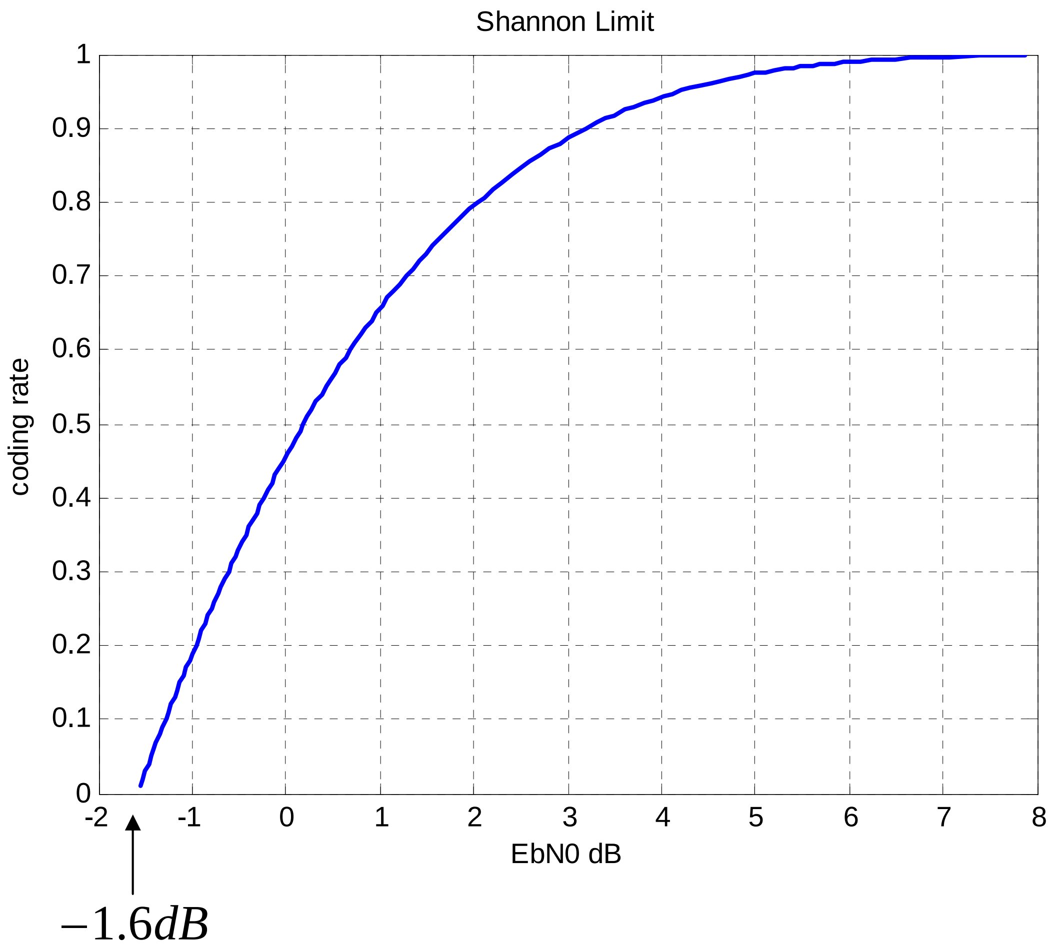
A. **Convolutional Encoder** with rate $r_c = 1/2$

B. **Block Code** **(65520, 61425)** **low density parity check**, with coding rate $r_c = \frac{61425}{65520} = \frac{15}{16}$



3. Minimum SNR Requirements

For a given Coding Rate, what is the minimum E_b / N_0 we can have?



Notice: $\dfrac{E_b}{N_0} > -1.6dB$ always!!!

2. Block Coding

2.1 Parameters of Block Codes

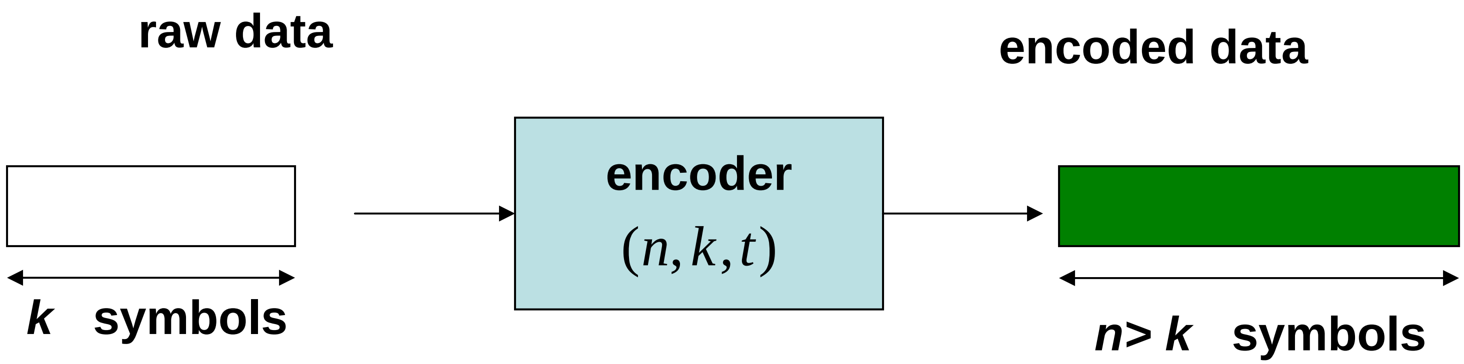
2.2 Probability of Error

2.3 Reed Solomon Codes (*non binary data)*

2.3 Simulink Implementation

1. Parameters of Block Codes

It encodes ***k*** blocks of data symbols into ***n*** blocks of encoded symbols:

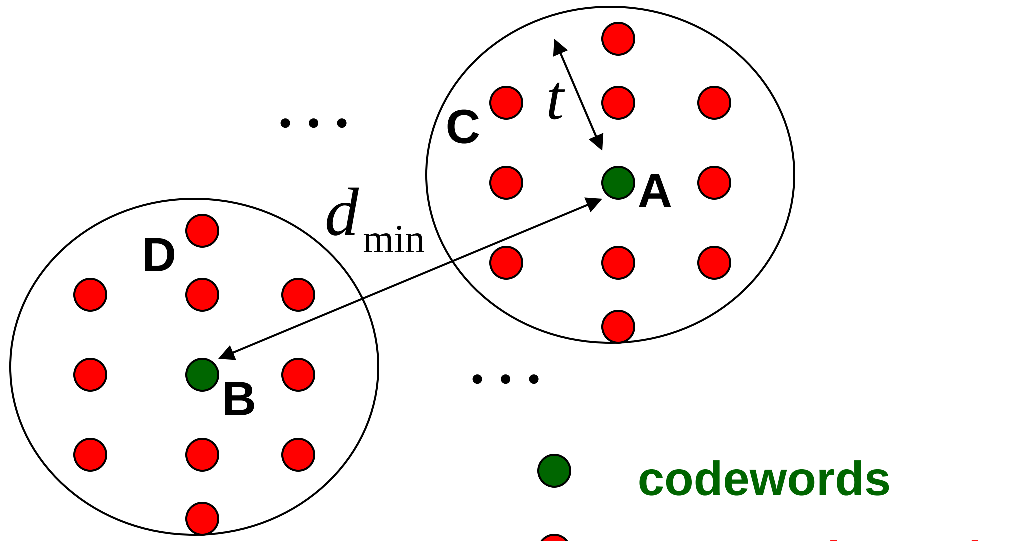


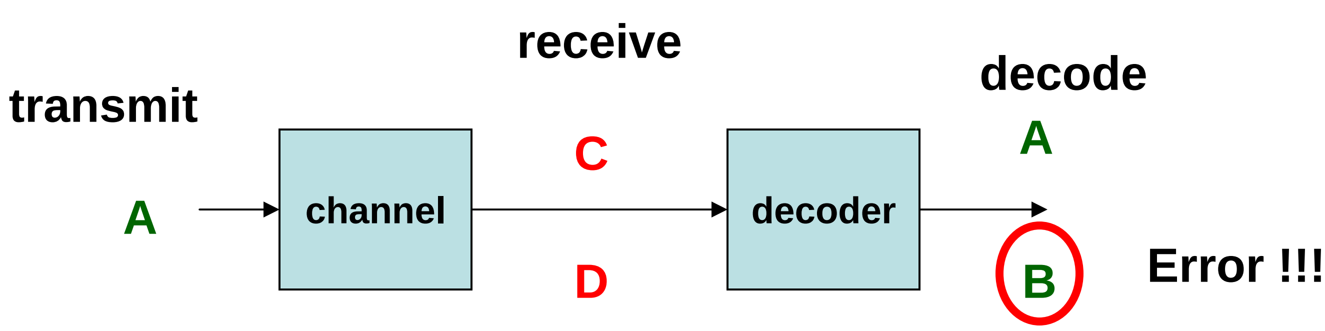
Code Rate $r_c = \dfrac{k}{n}$

<u>Parameters</u>: (n, k, t)

$$t \leq floor\left(\frac{d_{\min} - 1}{2}\right)$$ Max. number of errors corrected

where $d_{\min} \leq n - k + 1$ Minimum distance between codewords





where $d(C, A) \leq t$

$d(D, A) > t$

2. Probability of Error

If the code corrects up to *t* errors, the probability that the wrong codeword is decoded is given by

$$P_e \le \sum_{j=t+1}^{n} \binom{n}{j} p^j (1-p)^{n-j} \le (2^k - 1)\left(4p(1-p)\right)^{d_{\min}/2}$$

"*n-j*" bits right

"*j*" bits wrong

with *p* the probability of error for one bit.

For BPSK, QPSK: $p = Q\left(\sqrt{2E_c / N_0}\right)$ $\quad E_c = \text{Energy of Coded Bit} = \frac{k}{n} E_b$

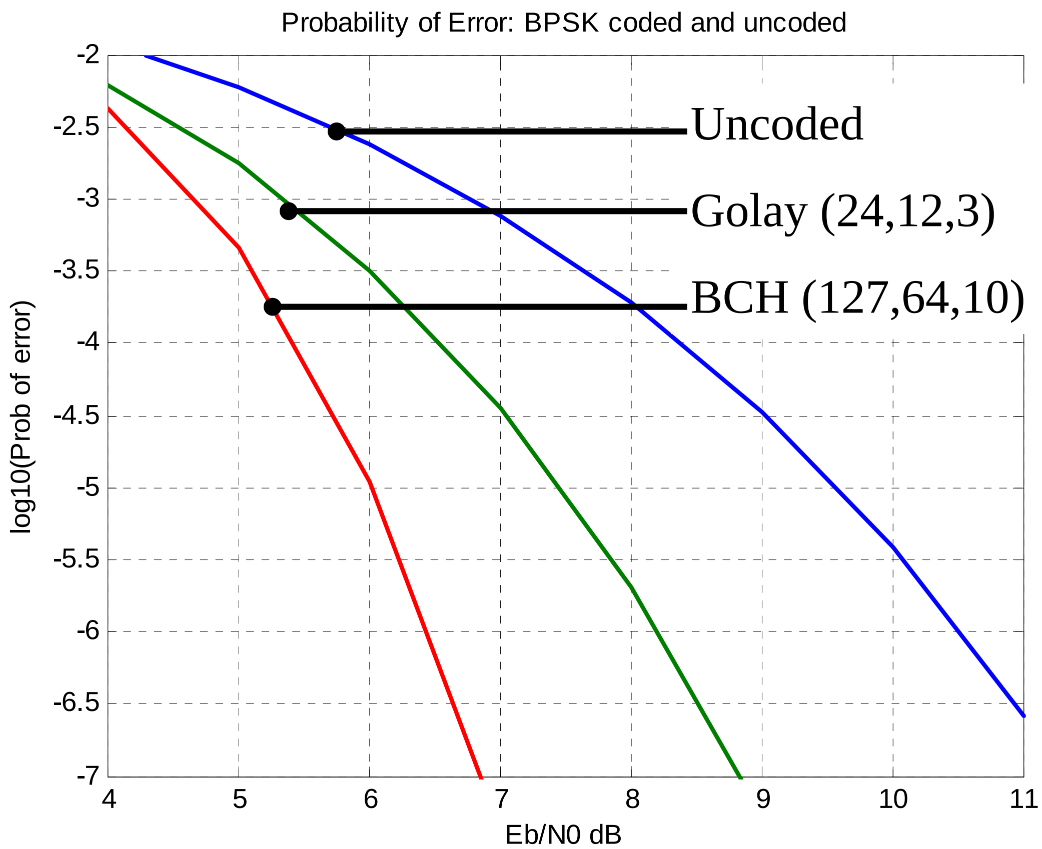
Bit Error Probability for HDD

$$P_b \approx \sum_{j=t+1}^{n} j \binom{n}{j} p^j (1-p)^{n-j} \approx \frac{P_e}{k}$$

codeword error

bits per codeword

Some Probabilities of Error for BPSK:



3. Reed Solomon Codes (non Binary Data)

$$(n, k, t)$$

$n = 2^m - 1$

coded symbols

$k < n$ uncoded symbols

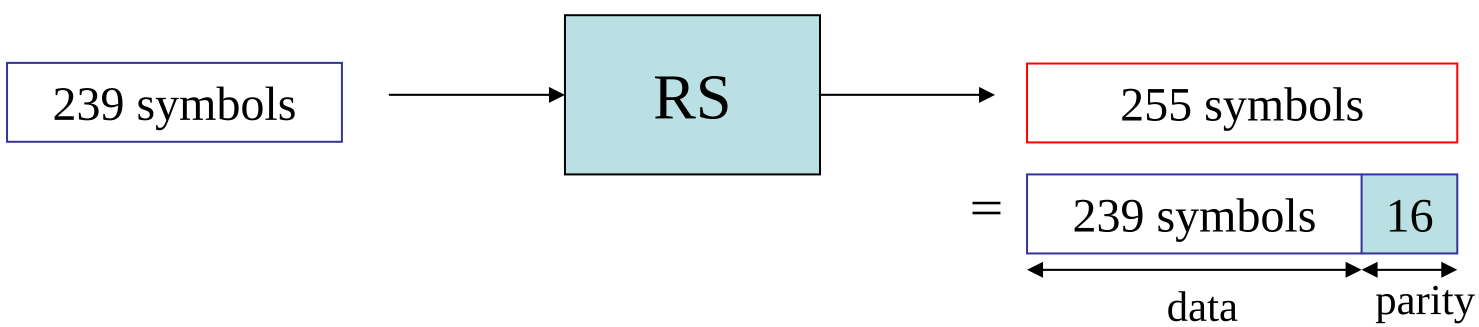
$t = floor\left(\dfrac{2^m - k}{2}\right)$

errors corrected

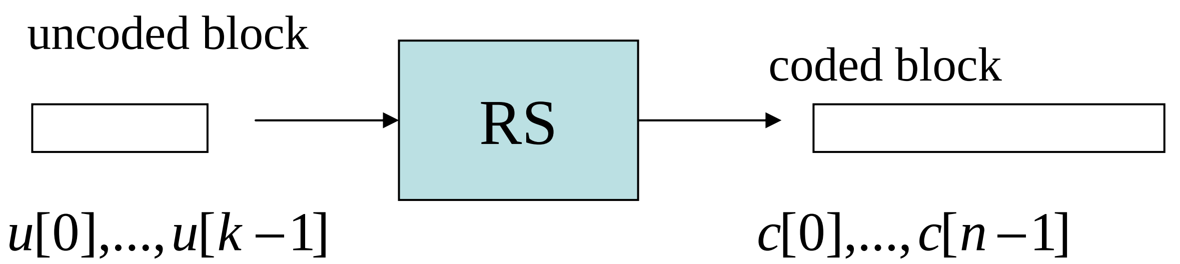
1 symbol = m bits

Good for Burst Error Correction at relatively high SNR

Example: (255,239,8)

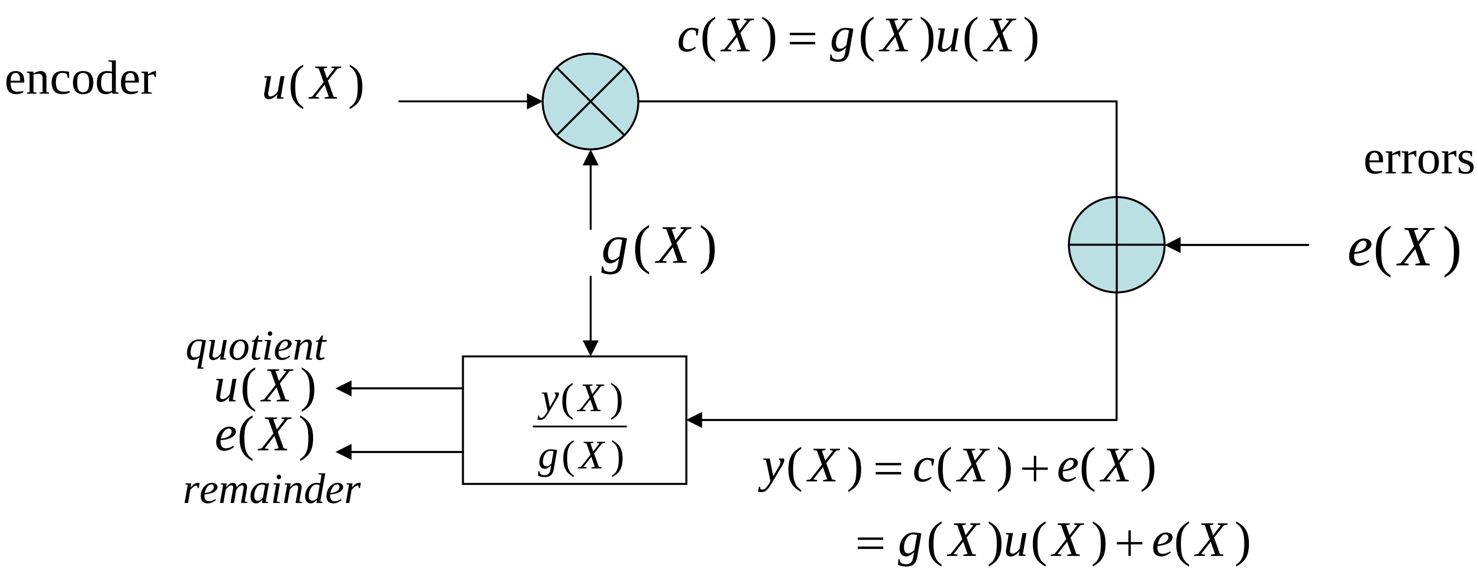


Based on polynomials:



$$u(X)=u[0]+u[1]X+\ldots+u[k-1]X^{k-1}$$

$$c(X)=c[0]+c[1]X+\ldots+c[n-1]X^{n-1}$$



3. Code Shortening and Puncturing

3.1 Code Shortening

3.2 Code Puncturing

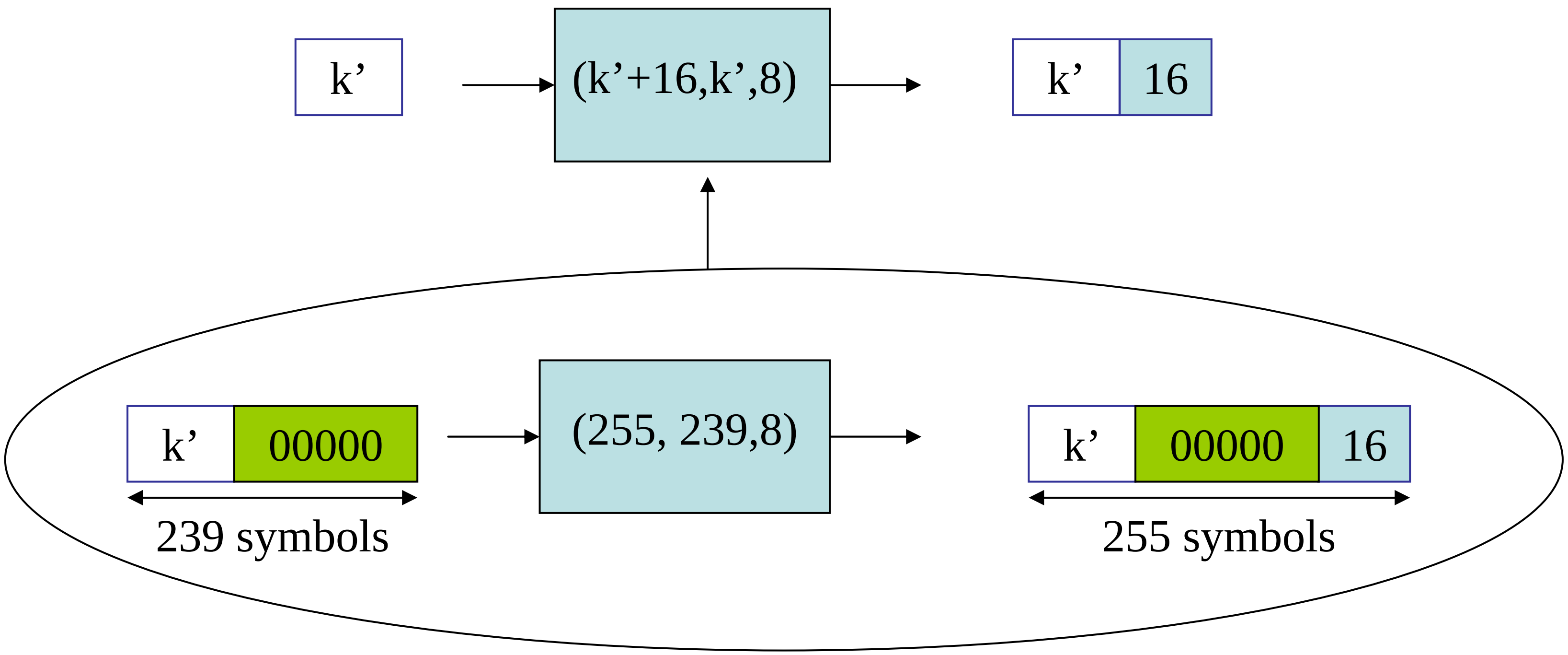
3.3 RS Coding in IEEE802.16

1. Code Shortening

Start with a block code, say RS (n=255, k=239, t=8).

We can generate a different code by **<u>Shortening</u>**:

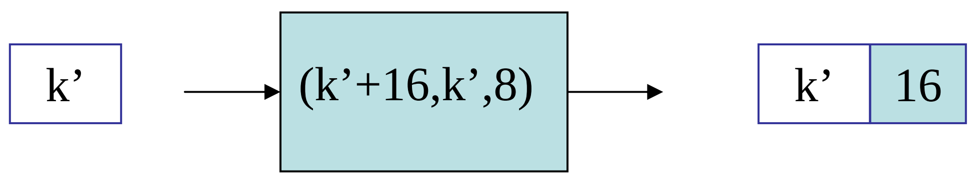
1. Shortened to k' data bytes by adding 239-k' null symbols
2. At the output eliminate the corresponding 239-k' terms



By shortening Reed Solomon (255,239,8) we obtain codes

$$(k'+16, k',8), \qquad 0 < k' \leq 239$$

All these codes correct up to 8 errors (bytes).



2. Code Puncturing

By puncturing we eliminate L parity bytes to obtain a code correcting 8-T errors. Therefore we determine L from

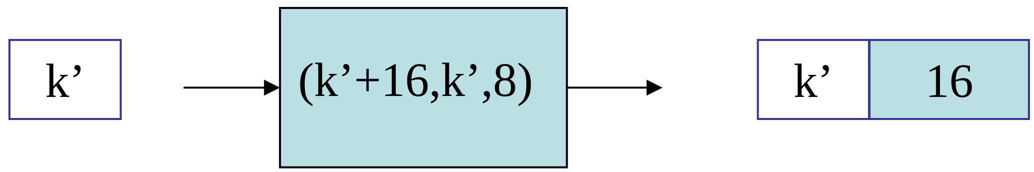
$$\frac{\overbrace{(N-k)}^{\text{parity bytes}}}{2} = 8 - T \quad \Rightarrow \quad \frac{16 - L}{2} = 8 - T \quad \Rightarrow L = 2T$$

Puncturing Reed Solomon Codes $(k'+16, k', 8)$, $0 < k' \le 239$

1. The last 16 bytes are parity bytes

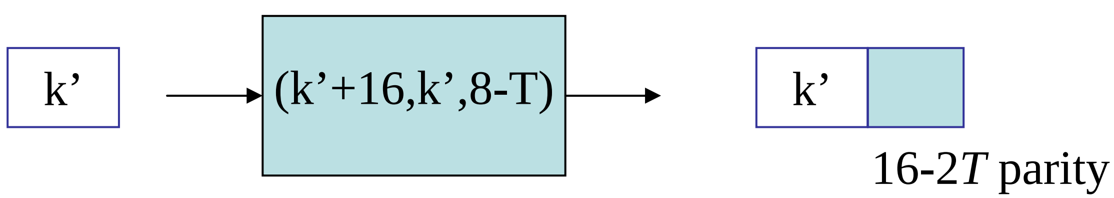
Code:

$(k'+16, k', 8)$



2. For a $(k'+16-2T, k', 8-T)$ code, correcting 8-T errors, just eliminate the first $2T$ bytes of the parity check.

$(k'+16-2T, k', 8-T)$



RS codes used in IEEE802.16: $\boxed{\left(k'+16-2T, k', 8-T\right)}$

(32,24,4) $\longleftarrow$ k'=24, T=4

(40,36,2) $\longleftarrow$ k'=36, T=6

(64,48,8) $\longleftarrow$ k'=48, T=0

(80,72,4) $\longleftarrow$ k'=72, T=4

(108,96,6) $\longleftarrow$ k'=96, T=2

(120,108,6) $\longleftarrow$ k'=108, T=2

4. Simulink Implementation of Block Codes

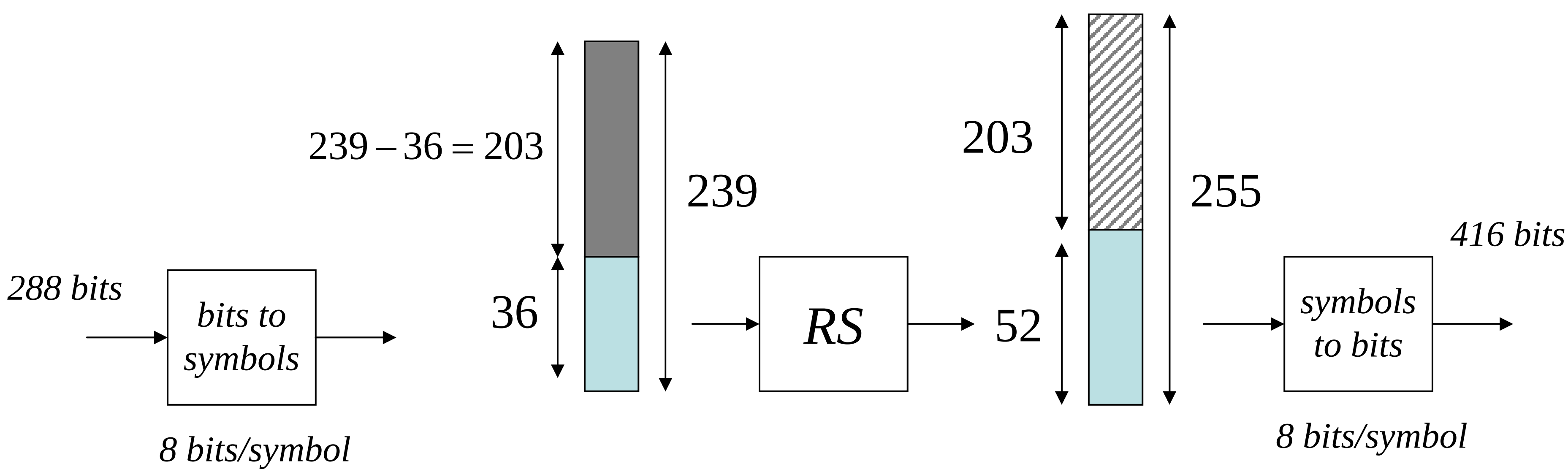
4.1 Shortened Codes

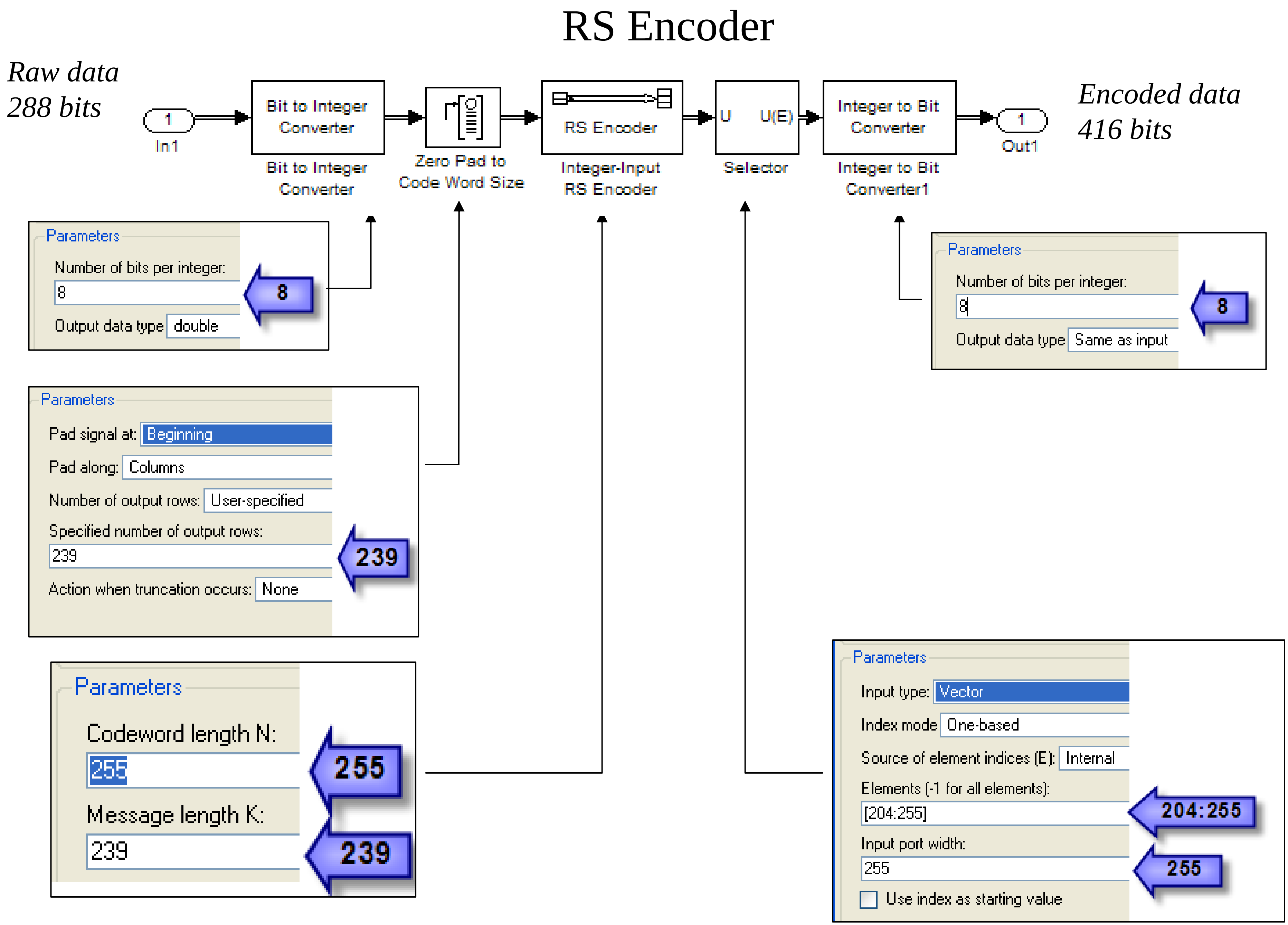
4.2 Punctured Codes

Simulink Implementation of Shortened RS Codes

Let's implement the code (52,36,8) by shortening

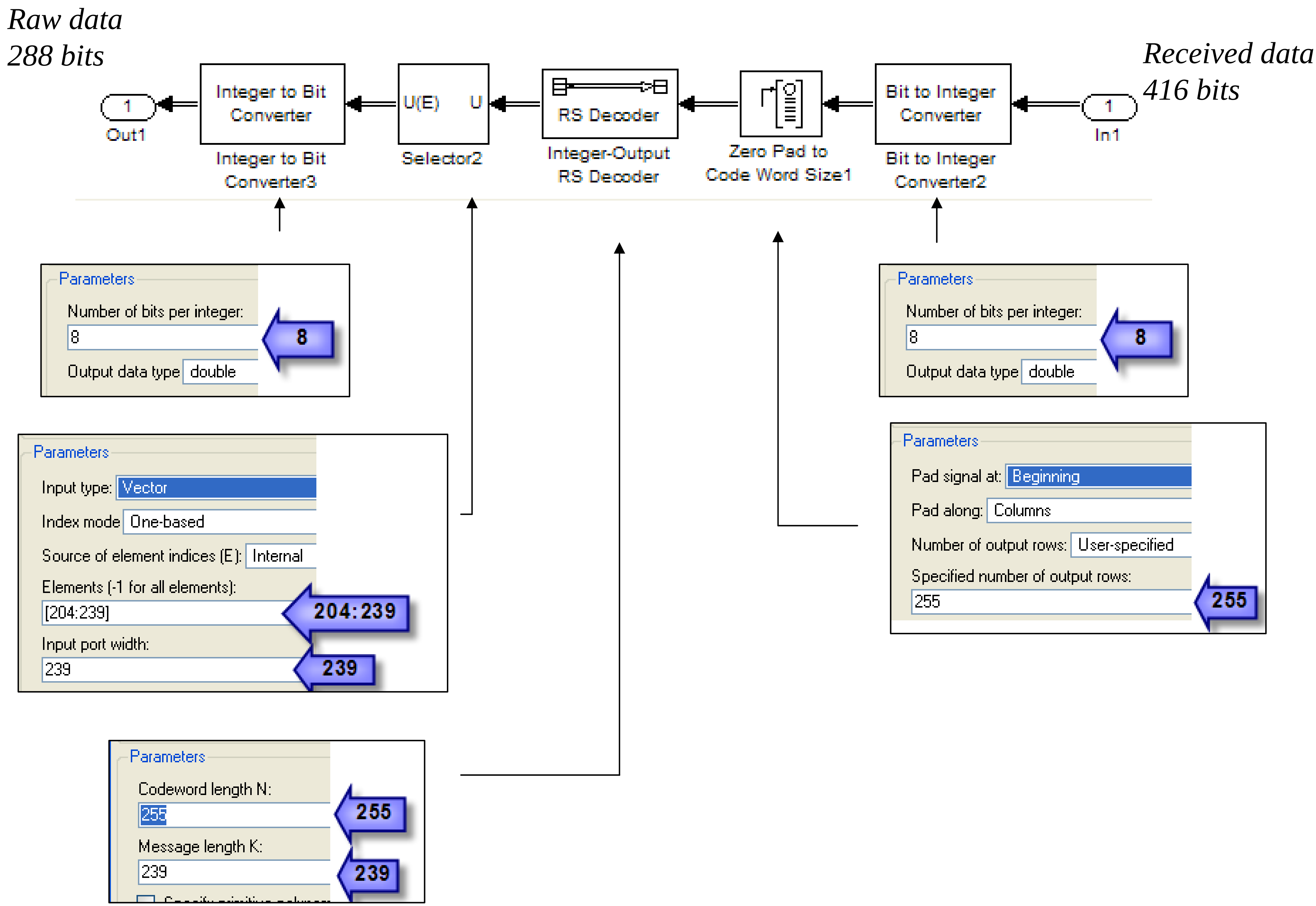
1. Shortened to k'=36 data *bytes* (=36x8=288 bits) by adding 239-k' null symbols at the beginning of the block
2. At the output eliminate the first 239-k' terms





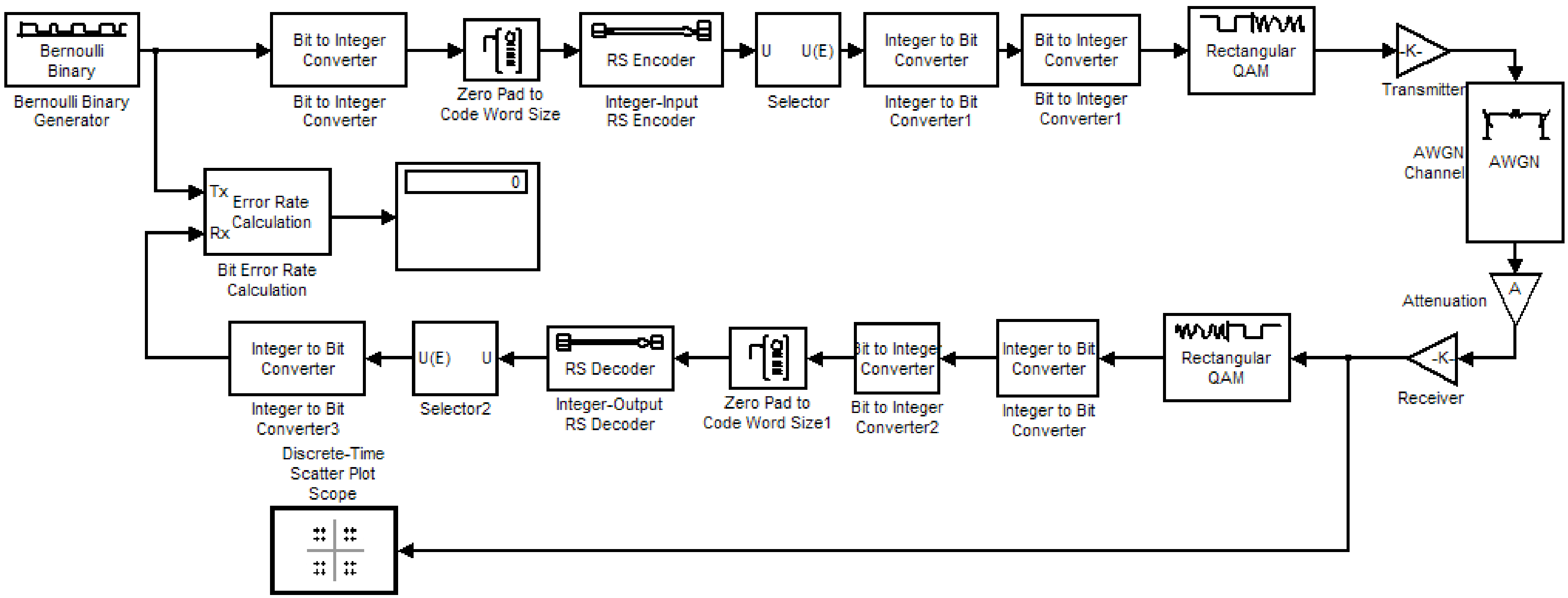
RS Encoder
Raw data
288 bits
1
In1
Bit to Integer Converter
Bit to Integer Converter
Zero Pad to Code Word Size
RS Encoder
Integer-Input RS Encoder
U
U(E)
Selector
Integer to Bit Converter
Integer to Bit Converter1
1
Out1
Encoded data
416 bits
Parameters
Number of bits per integer:
8
Output data type double
8
Parameters
Pad signal at: Beginning
Pad along: Columns
Number of output rows: User-specified
Specified number of output rows:
239
Action when truncation occurs: None
239
Parameters
Codeword length N:
255
Message length K:
239
255
239
Parameters
Number of bits per integer:
8
Output data type Same as input
8
Parameters
Input type: Vector
Index mode One-based
Source of element indices (E): Internal
Elements (-1 for all elements):
[204:255]
Input port width:
255
Use index as starting value
204:255
255

RS Decoder

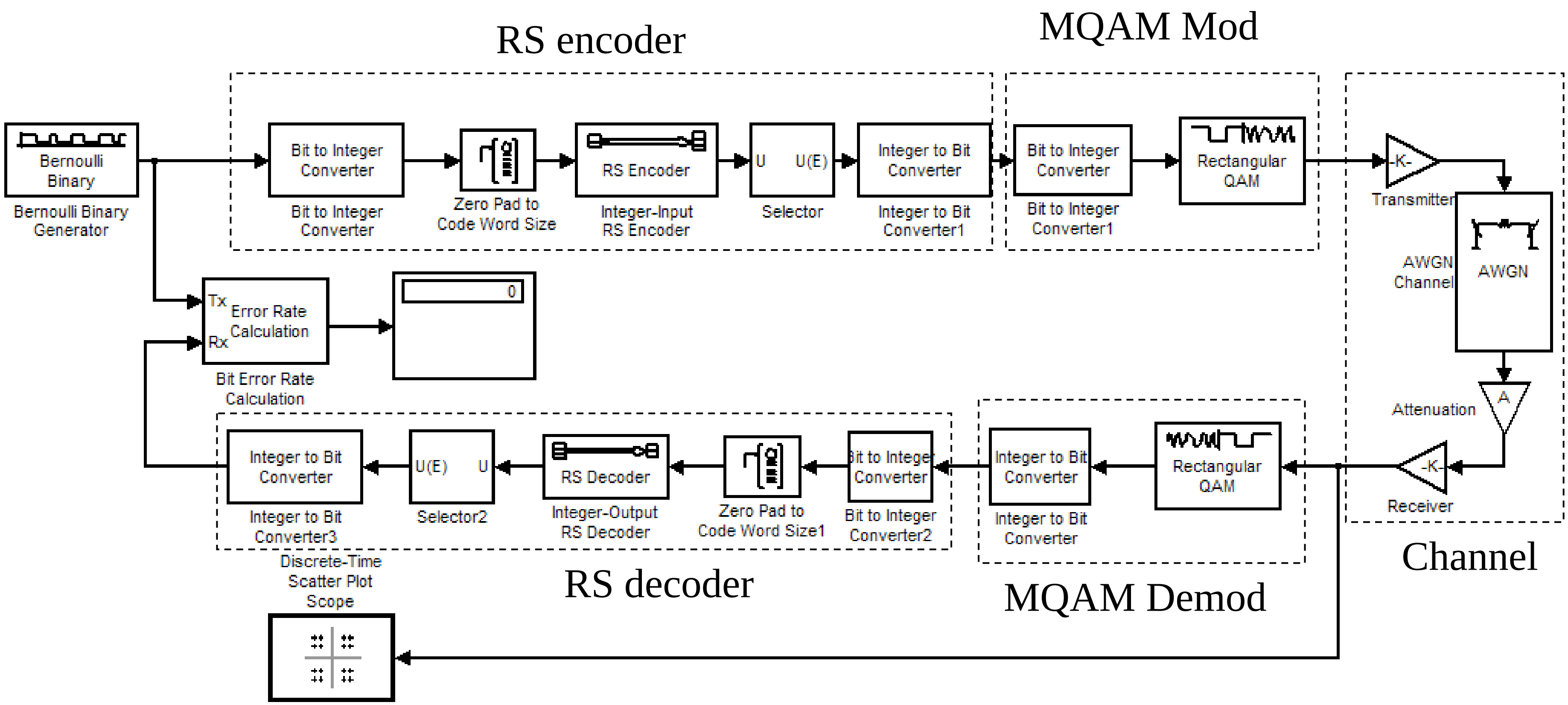


Raw data
288 bits
Received data
416 bits
1
Out1
Integer to Bit Converter
Integer to Bit Converter3
U(E) U
Selector2
RS Decoder
Integer-Output RS Decoder
Zero Pad to Code Word Size1
Bit to Integer Converter
Bit to Integer Converter2
1
In1
Parameters
Number of bits per integer:
8
Output data type double
8
Parameters
Input type: Vector
Index mode One-based
Source of element indices (E): Internal
Elements (-1 for all elements):
[204:239]
204:239
Input port width:
239
239
Parameters
Codeword length N:
255
255
Message length K:
239
239
Parameters
Number of bits per integer:
8
Output data type double
8
Parameters
Pad signal at: Beginning
Pad along: Columns
Number of output rows: User-specified
Specified number of output rows:
255
255

Put Everything together:



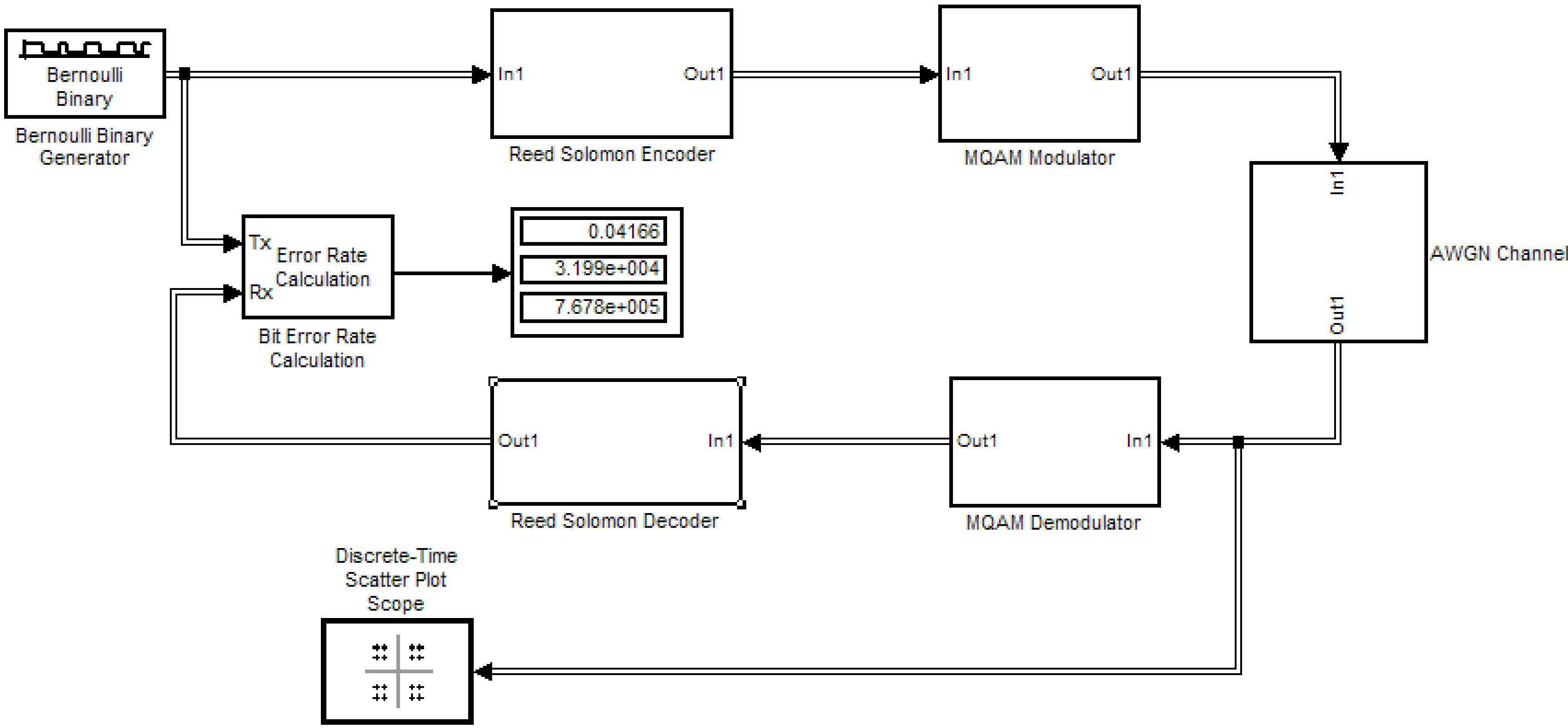
It is easier to create subsystems:



Select each group of blocks and use

Edit > Create Subsystem

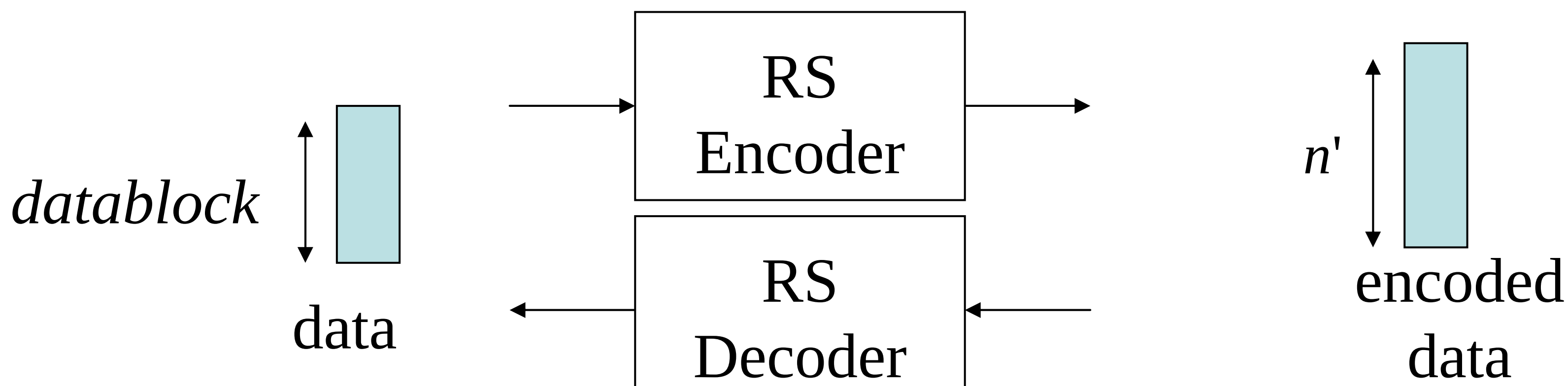
It cannot be undone!


Bernoulli
Binary
Bernoulli Binary
Generator
In1
Out1
Reed Solomon Encoder
In1
Out1
MQAM Modulator
In1
Out1
AWGN Channel
Tx
Error Rate
Calculation
Rx
Bit Error Rate
Calculation
0.04166
3.199e+004
7.678e+005
Out1
In1
Reed Solomon Decoder
Out1
In1
MQAM Demodulator
Discrete-Time
Scatter Plot
Scope

Masked Subsystems

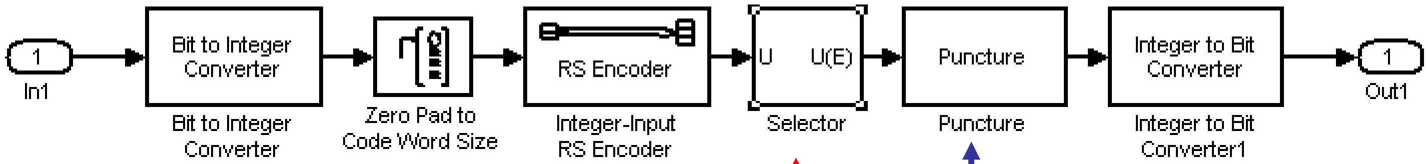
In Simulink we can create customized blocks, where we can enter parameters values.

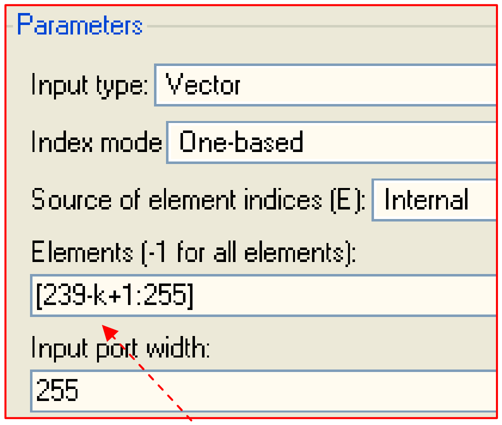
Example: Reed Solomon Encoder and Decoder



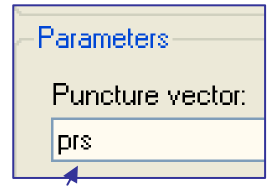
RS Encoder:

1. edit





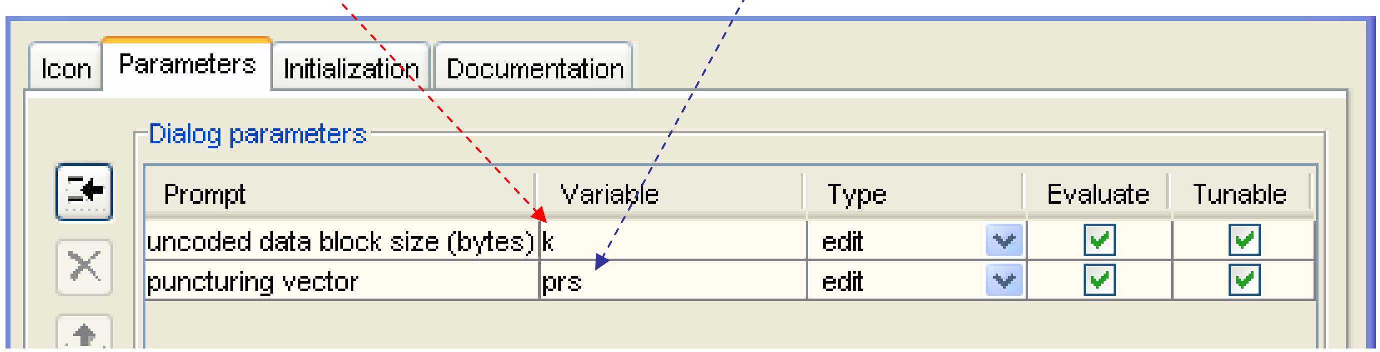
[239-k+1:255]



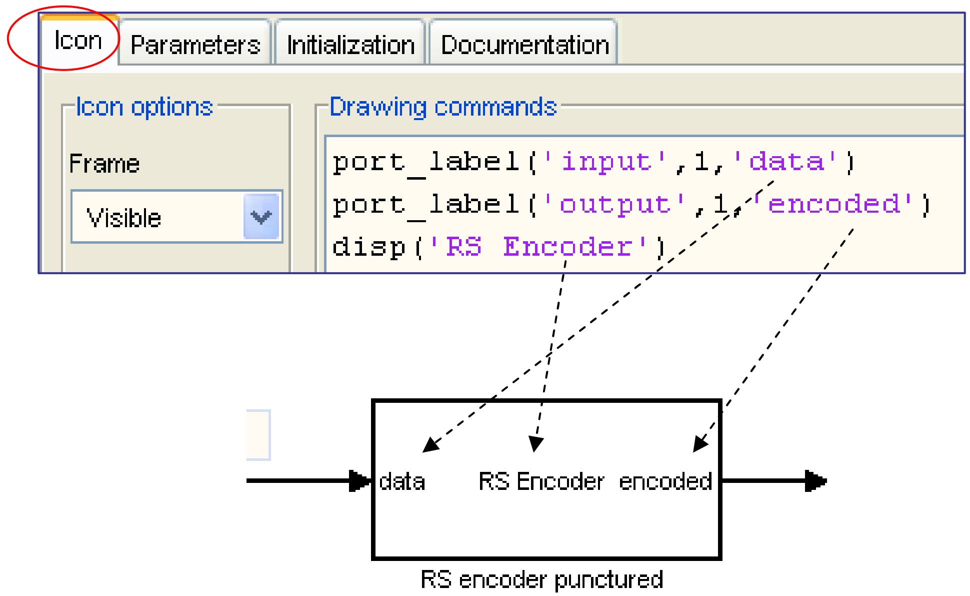
2. right click on subsystem and select *Mask Subsystem*

3. Edit Mask:

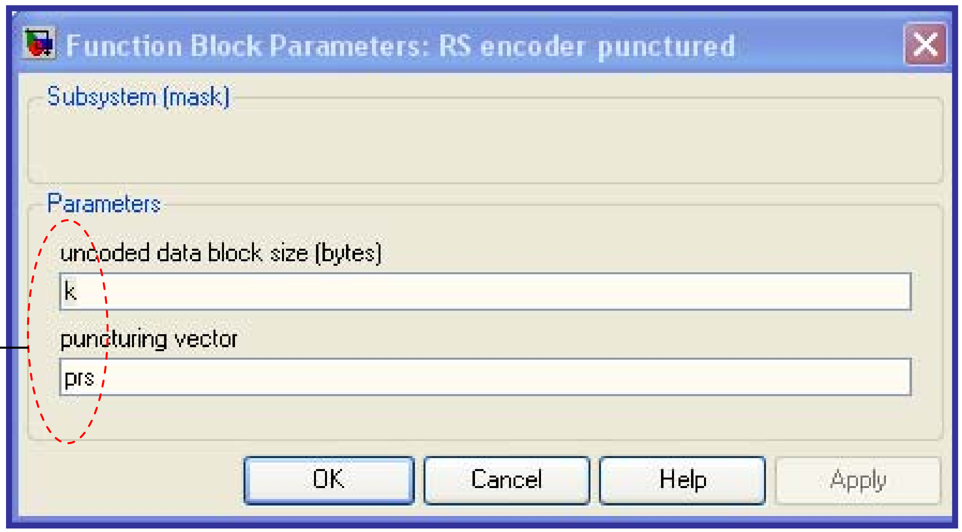
add parameters



You can customize the icon

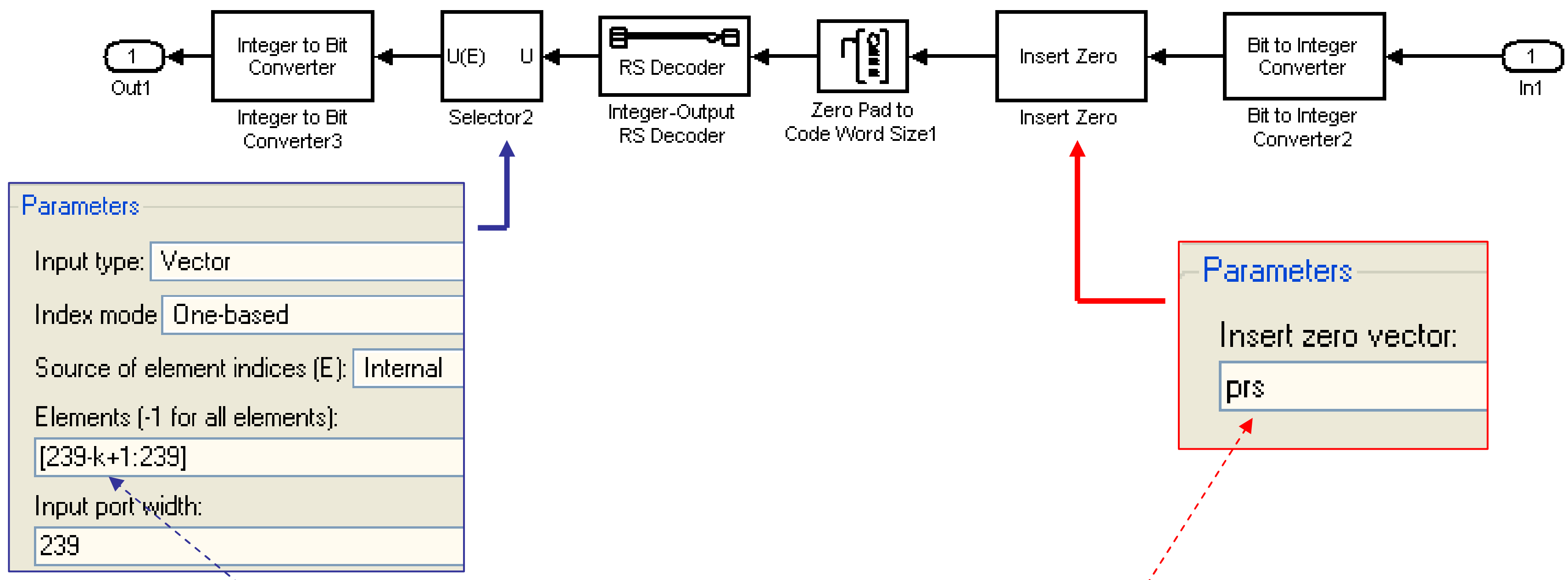


When you "click" on the icon, you get the following:



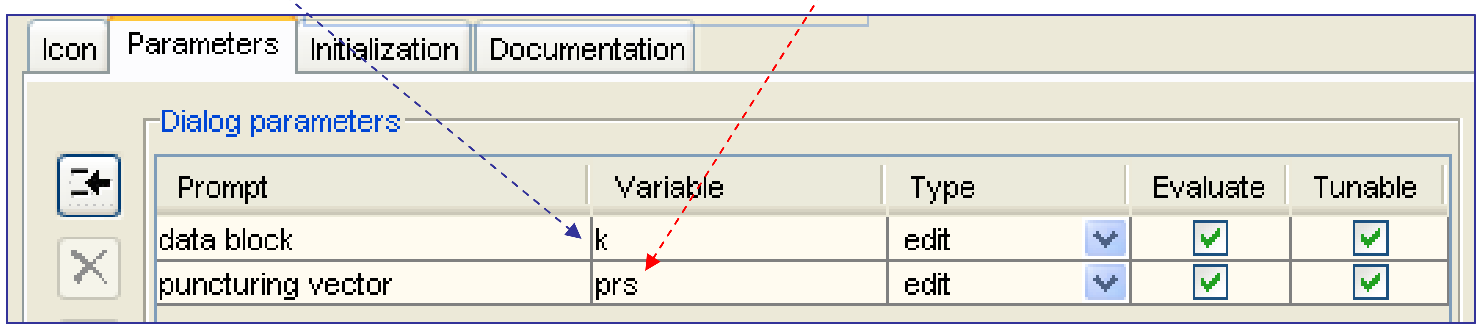
the two parameters from previous slide

RS Decoder:



2. right click on subsystem and select *Mask Subsystem*

3. Edit Mask:

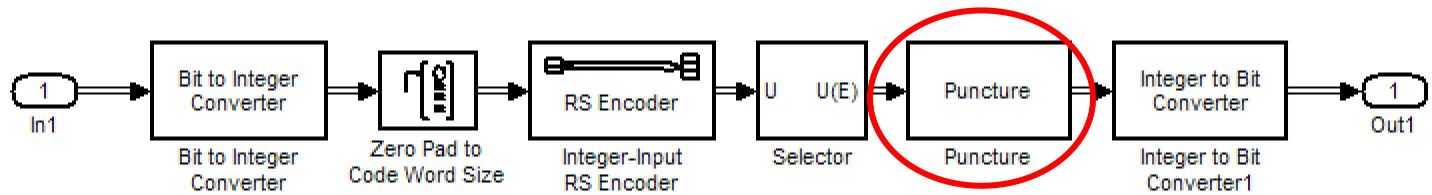


Simulink Implementation of (40,36,2).

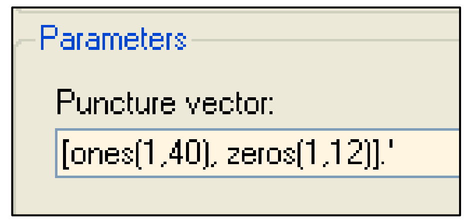
1. Start with the previous code (52,36,8);
2. Puncture it by eliminating $2T$=12 (ie T=6) bytes in the parity

RS (40,36,2) encoder

Sequence Operations>Puncture



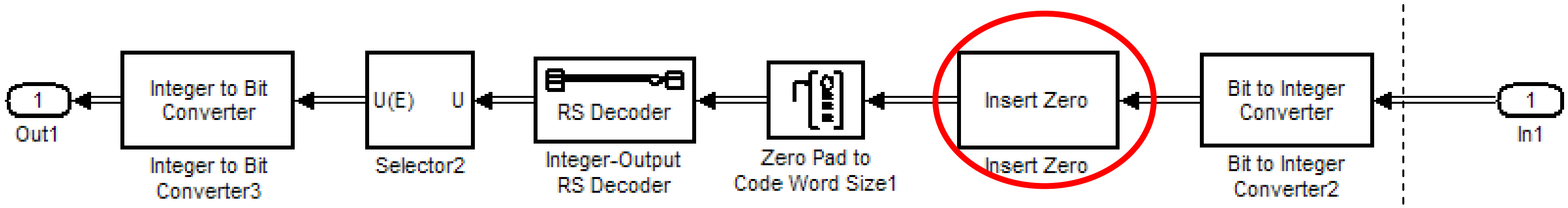
Eliminate last 12 bytes



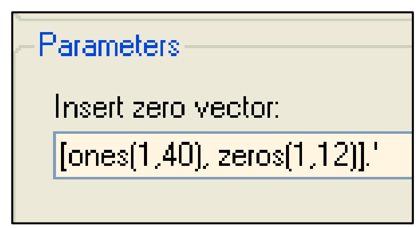
same vector

RS (40,36,2) decoder

Sequence Operations>Insert Zero



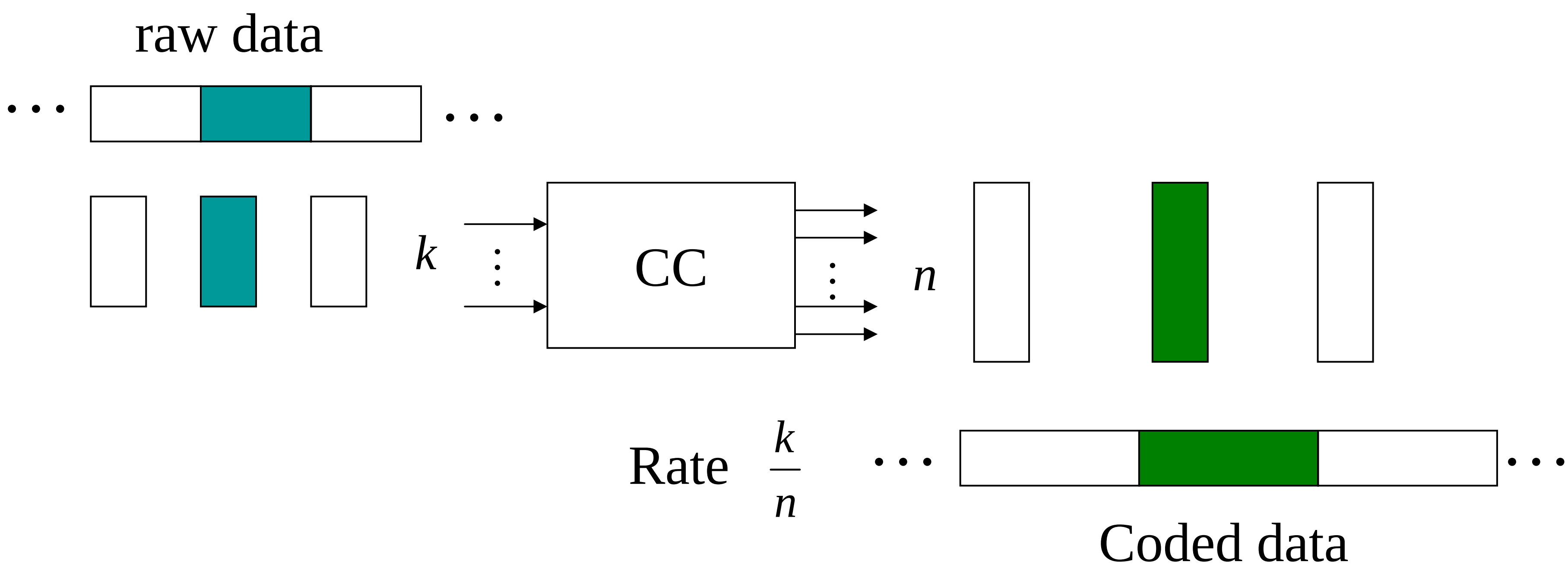
Insert 12 zeros



Convolutional Encoders

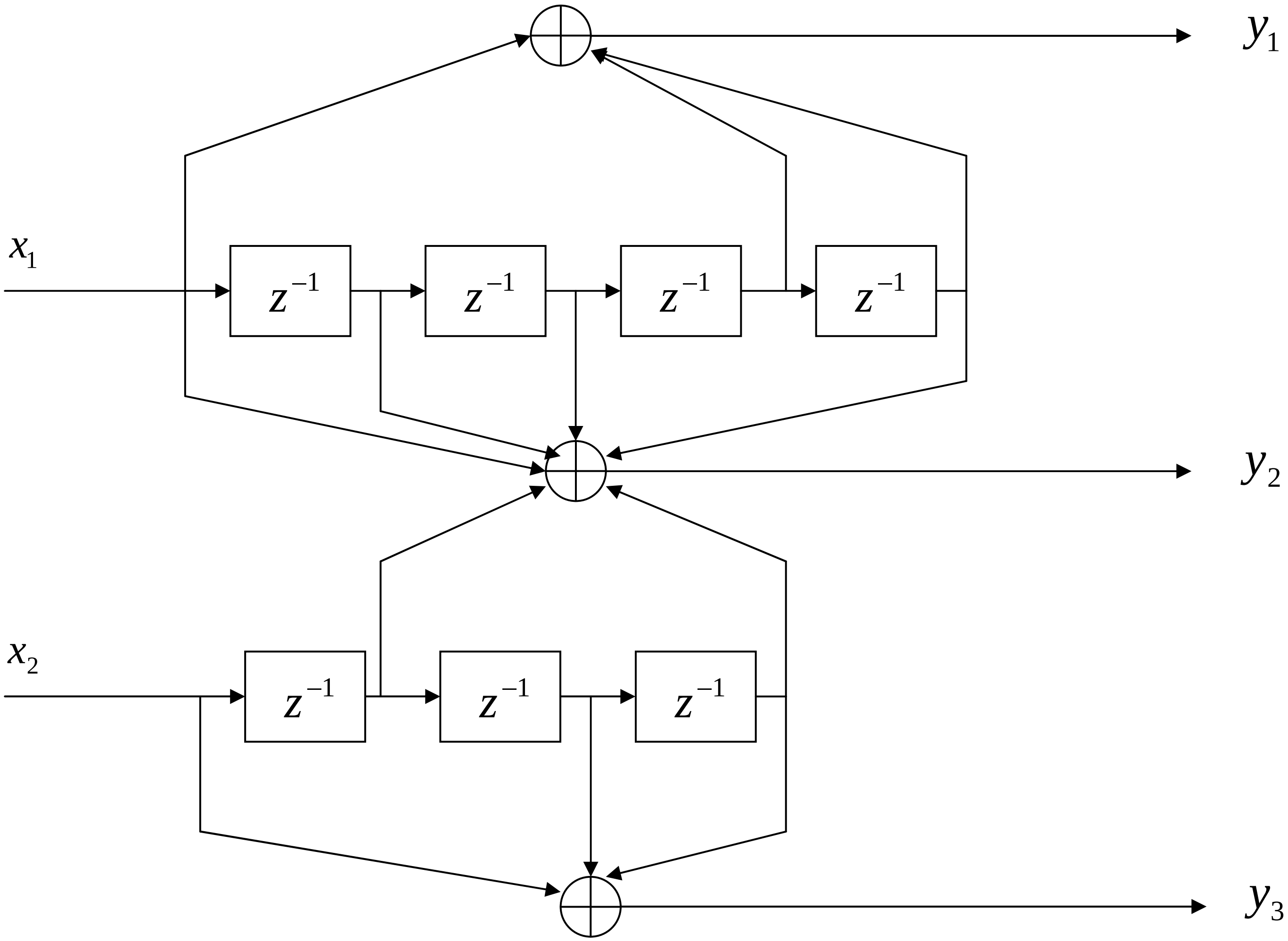
1. ***Definition of Convolutional Encoders***
2. ***Puncturing***
3. ***Simulink Implementation***

Convolutional Encoders



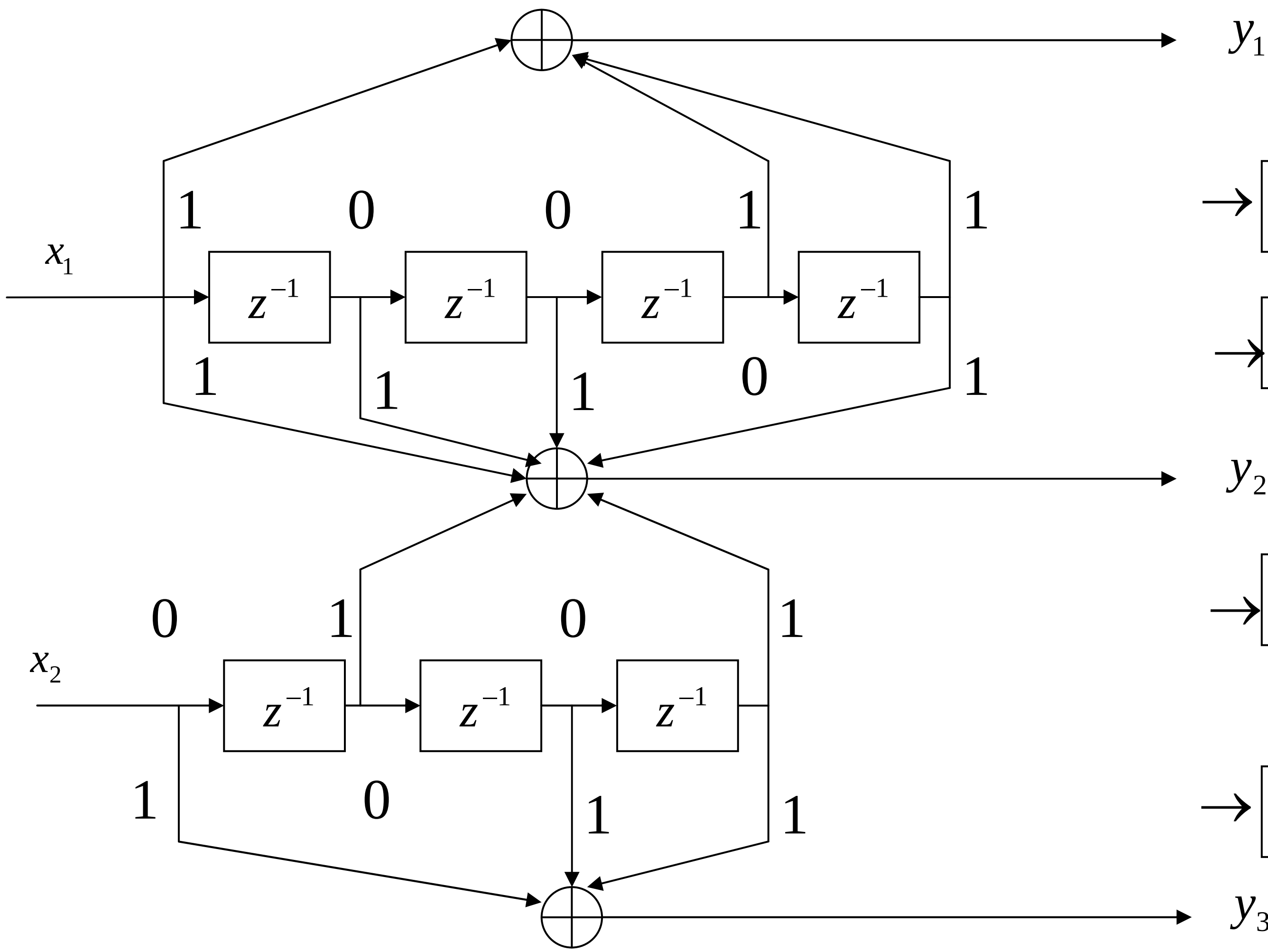
It continuously updates k data symbols into $n>k$ coded symbols.

Example: a 2/3 encoder



Polynomial description:

$$\begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$



$\rightarrow \boxed{a_{11} = (10)_8 (011)_8 = 23}$

$\rightarrow \boxed{a_{12} = (11)_8 (101)_8 = 35}$

$\rightarrow \boxed{a_{22} = (0)_8 (101)_8 = 5}$

$\rightarrow \boxed{a_{23} = (1)_8 (011)_8 = 13}$

Therefore the code is described by the matrix (all entries are octal)

no connection $x_1 \rightarrow y_3$

$$\begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 23 & 35 & 0 \\ 0 & 5 & 13 \end{bmatrix}$$

no connection $x_2 \rightarrow y_1$

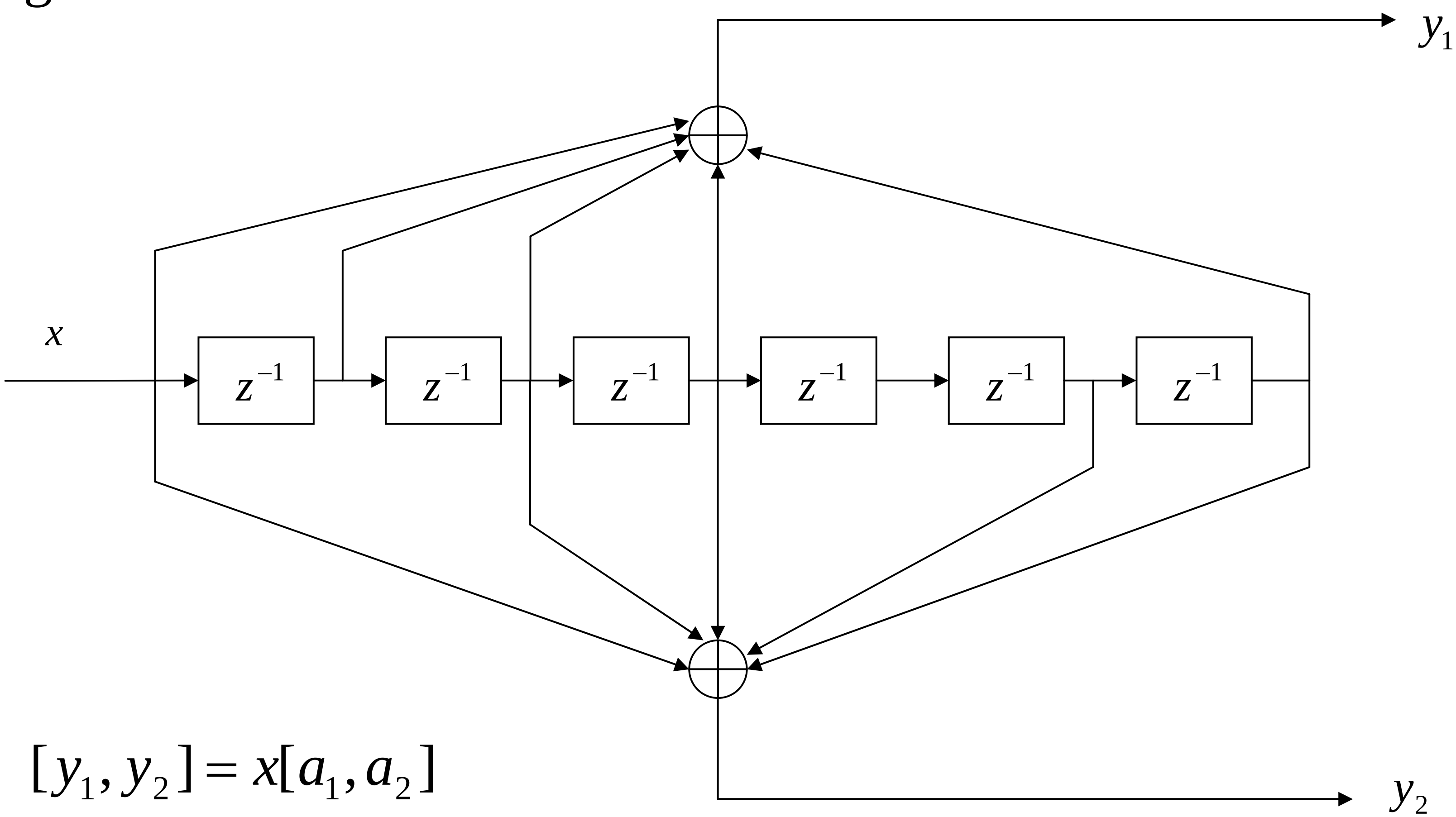
The parameters are then:

CONSTRAINED_LENGTH = [4,3]

CODEGENERATOR=[23,35,0; 0,5,13]

which call the function “poly2trellis”

In IEEE802.16e the Convolutinal Code has rate ½ and constrained length 7:



$$[y_1, y_2] = x[a_1, a_2]$$

$$a_1 = (1)_8 (111)_8 (001)_8 = 171$$

$$a_2 = (1)_8 (011)_8 (011)_8 = 133$$

constrained length = 7

Note: in some books (as in [Costello]) the binary coefficients are determined in reverse order.

Example: the code in the previous page is defined by he polynomials (begin counting from the right) as

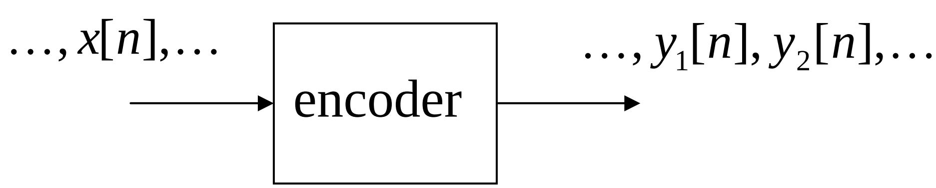
$$g_1 = (1)_8 (001)_8 (111)_8 = 117$$
$$g_2 = (1)_8 (011)_8 (011)_8 = 155$$

Punctured Codes

From an (k,n) convolutional code make a code with higher rate (less correcting capabilities), by periodically eliminating output bits.

Example in IEEE 802.16e:

The code [171,133] seen before is a (1,2) code:



Rate 2/3:

$$\ldots, x[n-1], x[n], \ldots \longrightarrow \boxed{\text{encoder}} \longrightarrow \ldots, y_1[n-1], y_2[n-1], \cancel{y_1[n]}, y_2[n], \ldots$$

We can represent it as a matrix with 2 rows:

$$P = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

Since:

$$\ldots, \overset{\boxed{1}}{y_1[n-1]}, \underset{\boxed{1}}{y_2[n-1]}, \overset{\boxed{0}}{y_1[n]}, \underset{\boxed{1}}{y_2[n]}, \ldots$$

Rate 3/4:

$$\ldots, x[n-2], x[n-1], x[n], \ldots \longrightarrow \boxed{\text{encoder}} \longrightarrow \ldots, y_1[n-2], y_2[n-2], \cancel{y_1[n-1]}, y_2[n-1], y_1[n], \cancel{y_2[n]}, \ldots$$

We can represent it as a matrix with 2 rows:

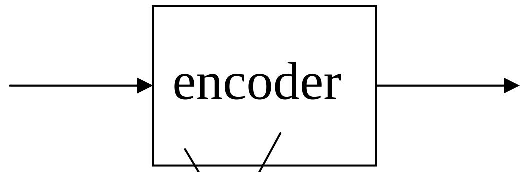
$$P = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Since:

$$\ldots, \overset{\boxed{1}}{y_1[n-2]}, \underset{\boxed{1}}{y_2[n-2]}, \overset{\boxed{0}}{\cancel{y_1[n-1]}}, \underset{\boxed{1}}{y_2[n-1]}, \overset{\boxed{1}}{y_1[n]}, \underset{\boxed{0}}{\cancel{y_2[n]}}, \ldots$$

Rate 5/6:

$$\ldots, x[n-4], x[n-3], x[n-2], x[n-1], x[n], \ldots$$



$$\ldots, y_1[n-4], y_2[n-4], \cancel{y_1[n-3]}, y_2[n-3], y_1[n-2], \cancel{y_2[n-2]}, \cancel{y_1[n-1]}, y_2[n-1], y_1[n], \cancel{y_2[n]}, \ldots$$

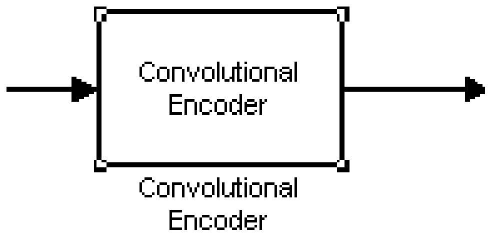
We can represent it as a matrix with 2 rows:

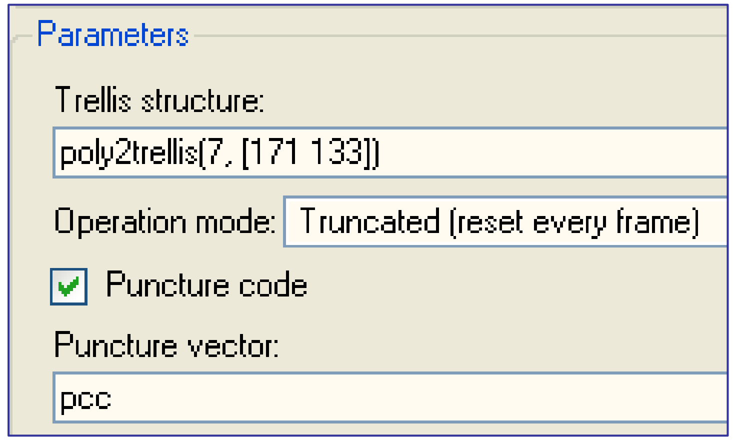
$$P = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

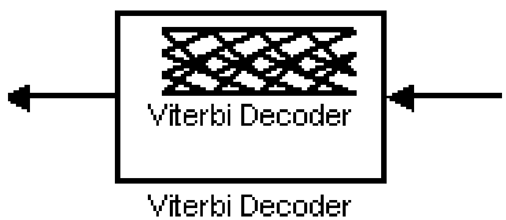
Since:

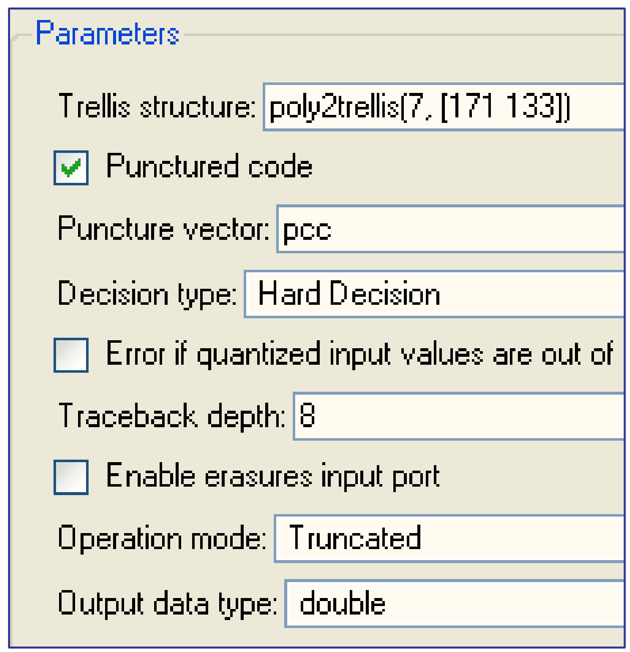
$$\ldots, \overset{\boxed{1}}{y_1[n-4]}, \underset{\boxed{1}}{y_2[n-4]}, \overset{\boxed{0}}{\cancel{y_1[n-3]}}, \underset{\boxed{1}}{y_2[n-3]}, \overset{\boxed{1}}{y_1[n-2]}, \underset{\boxed{0}}{\cancel{y_2[n-2]}}, \overset{\boxed{0}}{\cancel{y_1[n-1]}}, \underset{\boxed{1}}{y_2[n-1]}, \overset{\boxed{1}}{y_1[n]}, \underset{\boxed{0}}{\cancel{y_2[n]}}, \ldots$$

Simulink Implementation



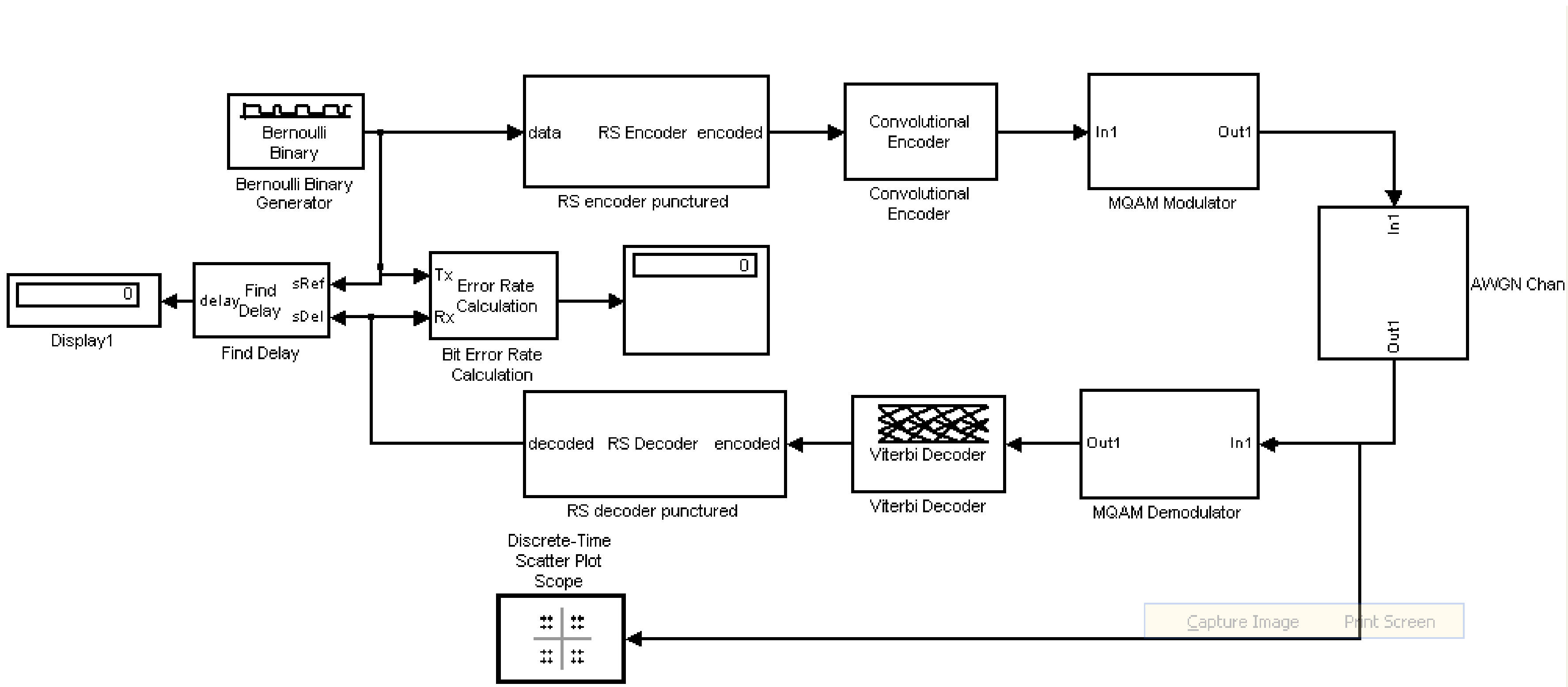






Implementation of Concatenated Codes using Simulink

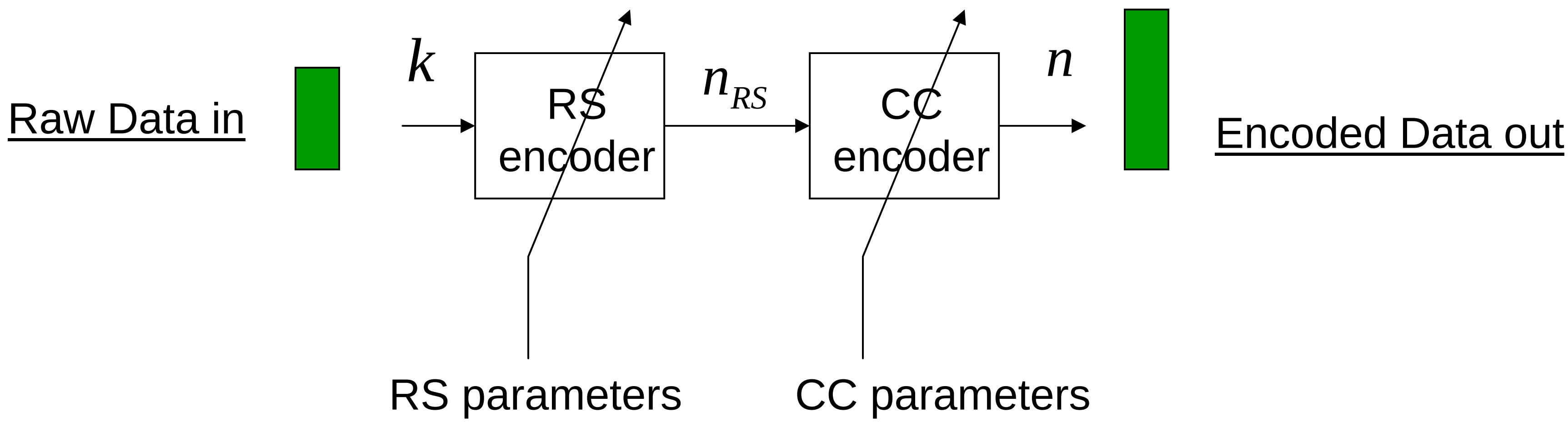
AWGN_RSCC_masked.mdl



Variable DataRates with Concatenated Codes in IEEE802.16

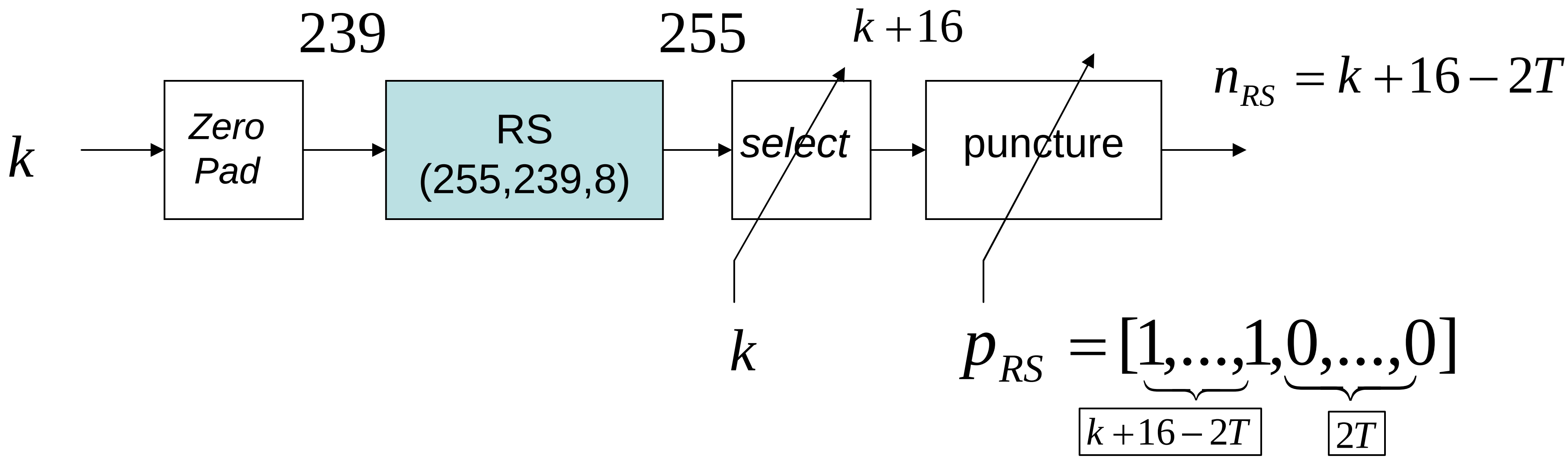
- It is highly desirable to be able to set the system to a number of different data and coding rates to adapt to channel conditions;
- IEEE802.16 achieves variable data rates by a combination of mechanisms:
    - coding (shortening, puncturing)
    - data block length (subchannelization)
- different rates have to be easily achieved by changing appropriate parameters without major reconfigurations.

Recall from previous units:

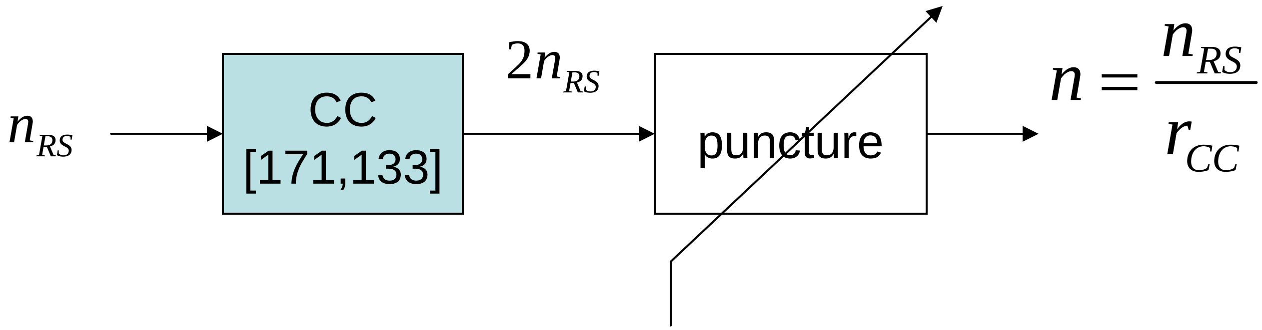


- encoder/decoder
- reduction
- puncturing

<u>RS Encoder</u> $\left(k+16-2T, k, 8-T\right)$



Convolutional Encoder with coding rate r_{CC}



$r_{CC} = 1/2, \quad P = [1,1]$

$r_{CC} = 2/3, \quad P = [1,1,0,1]$

$r_{CC} = 3/4, \quad P = [1,1,0,1,1,0]$

$r_{CC} = 5/6, \quad P = [1,1,0,1,1,0,0,1,1,0]$

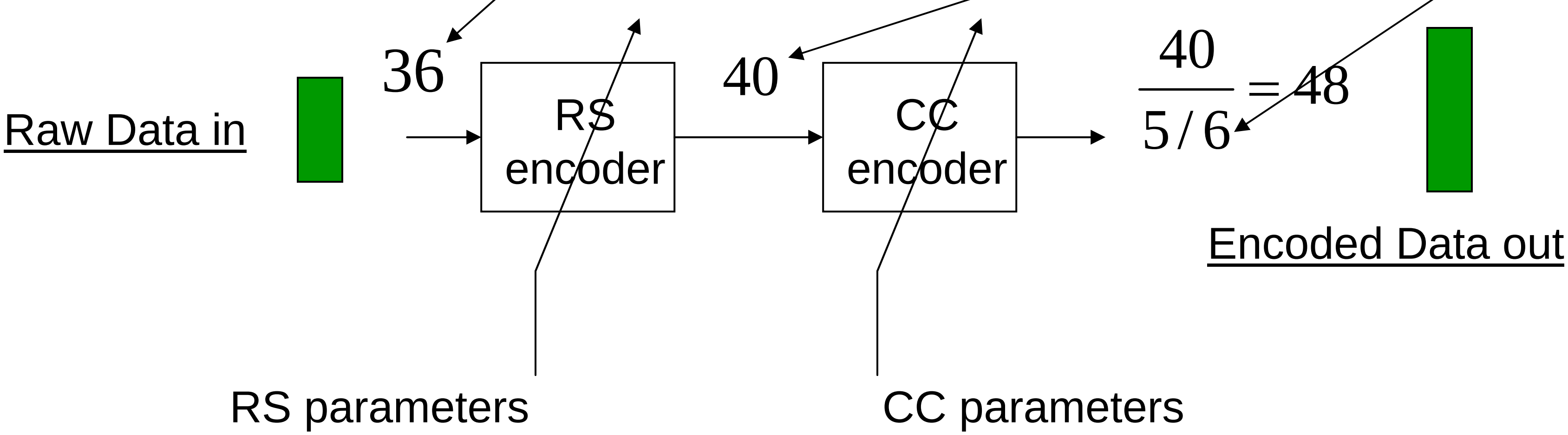
By combining the two codes we obtain a number of data rates (section 8.3.3.2 of the standard):

Rate ID	M-QAM	Data block *k* (bytes)	Coded block *n* (bytes)	Coding rate *k/n*	RS Code	CC coding rate
0	2	12	24	1/2	(12,12,0)	1/2
1	4	24	48	1/2	(32,24,4)	2/3
2	4	36	48	3/4	(40,36,2)	5/6
3	16	48	96	1/2	(64,48,8)	2/3
4	16	72	96	3/4	(80,72,4)	5/6
5	64	96	144	2/3	(108,96,6)	3/4
6	64	108	144	3/4	(120,108,6)	5/6

Example:

take Rate_ID=2

Rate ID	M-QAM	Data block *k* (bytes)	Coded block *n* (bytes)	Coding rate *k/n*	RS Code	CC coding rate
2	4	36	48	3/4	(40,36,2)	5/6



$$k = 36$$

$$p_{RS} = [\underbrace{1,\dots,1}_{\boxed{36+16-2\times 6=40}},\underbrace{0,\dots,0}_{\boxed{2\times 6=12}}]$$

$$p_{CC} = [1,1,0,1,1,0,0,1,1,0]$$

This yields a table of parameters for each “Rate_ID”:

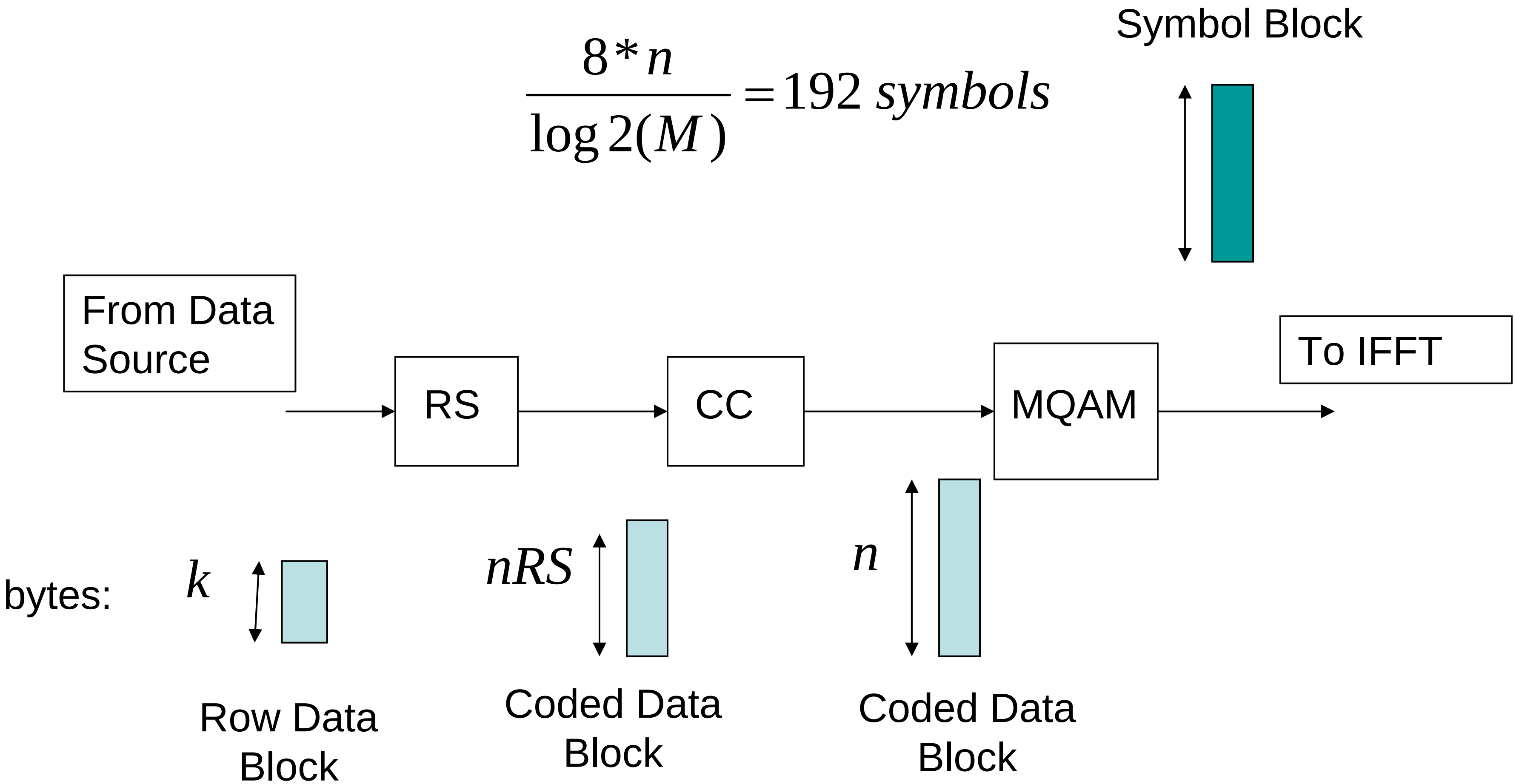
Rate ID	M-QAM	Data block k (bytes)	p_{RS}	p_{CC}	Coded block n (bytes)	CC coding rate
0	2	12	No RS	[1,1]	24	1/2
1	4	24	[ones(1,32),zeros(1,8)]	[1,1,0,1]	48	2/3
2	4	36	[ones(1,40),zeros(1,12)]	[1,1,0,1,1,0,0,1,1,0]	48	5/6
3	16	48	1	[1,1,0,1]	96	2/3
4	16	72	[ones(1,80),zeros(1,8)]	[1,1,0,1,1,0,0,1,1,0]	96	5/6
5	64	96	[ones(1,108),zeros(1,4)]	[1,1,0,1,1,0]	144	3/4
6	64	108	[ones(1,120),zeros(1,4)]	[1,1,0,1,1,0,0,1,1,0]	144	5/6

All parameters can be computed from the matlab program **coding.m**

```
function [M, k, prs, pcc]=coding(rate_ID);
% [M, k, prs, pcc]=coding(rate_ID)
% M= MQAM modulation
% k=input block size (bytes)
% prs=puncturing vector for RS encoder
% pcc=puncturing vector for CC encoder
% rate_ID=index to coding scheme 0-6 as in the standard

% Build structure:
% MQAM Modulation
coding_parameters(1).MQAM=2;
...
coding_parameters(7).MQAM=64;

% Input block size (in bytes)
coding_parameters(1).k=12;
...
coding_parameters(7).k=108;

% RS puncturing vector
coding_parameters(1).prs=-1; % no RS
coding_parameters(2).prs=[ones(1,32),zeros(1,8)];
...
coding_parameters(7).prs=[ones(1,120),zeros(1,4)];

% CC puncturing vector
coding_parameters(1).pcc=[1,1];
...
coding_parameters(7).pcc=[1,1,0,1,1,0,0,1,1,0];

% select data for given Rate_ID
n=rate_ID+1;
M=coding_parameters(n).MQAM;
k=coding_parameters(n).k;
prs=coding_parameters(n).prs';
pcc=coding_parameters(n).pcc';
```

The Coding Rates and the MQAM modulation parameters are designed in such a way that, in all cases

$$\frac{8*n}{\log 2(M)} = 192\ symbols$$



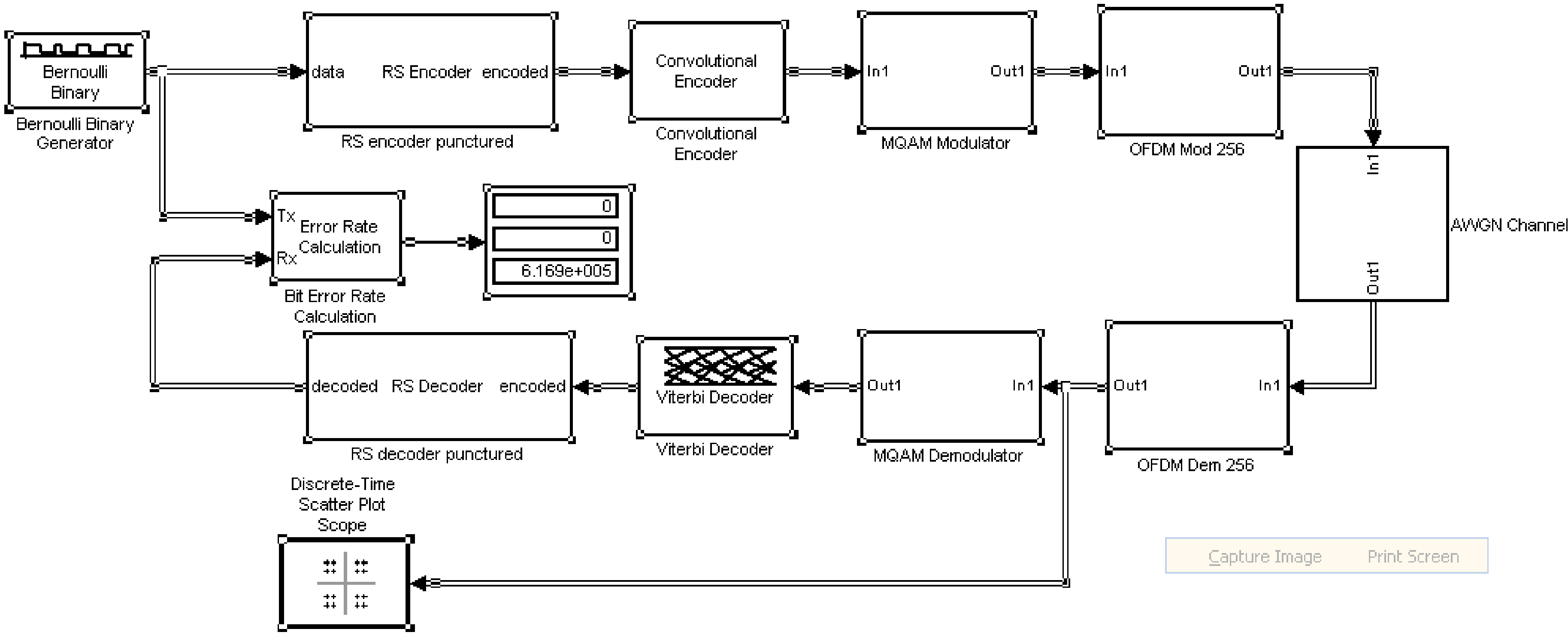
In all cases of NFFT=256,512,1024,2048 the number of data symbols transmitted is a multiple of 192.

Recall this table:

FFT size	256	128	512	1024	2048
N_used	200	108	426	850	1702
N_nulls	56	20	86	174	346
N_pilots	8	12	42	82	166
N_data	192	96	384	768	1536
			2x192	4x192	8x192

Put Everything together for IEEE802.16 2004 with AWGN Channel

WiMax256.mdl



IEEE802.16 Implementation

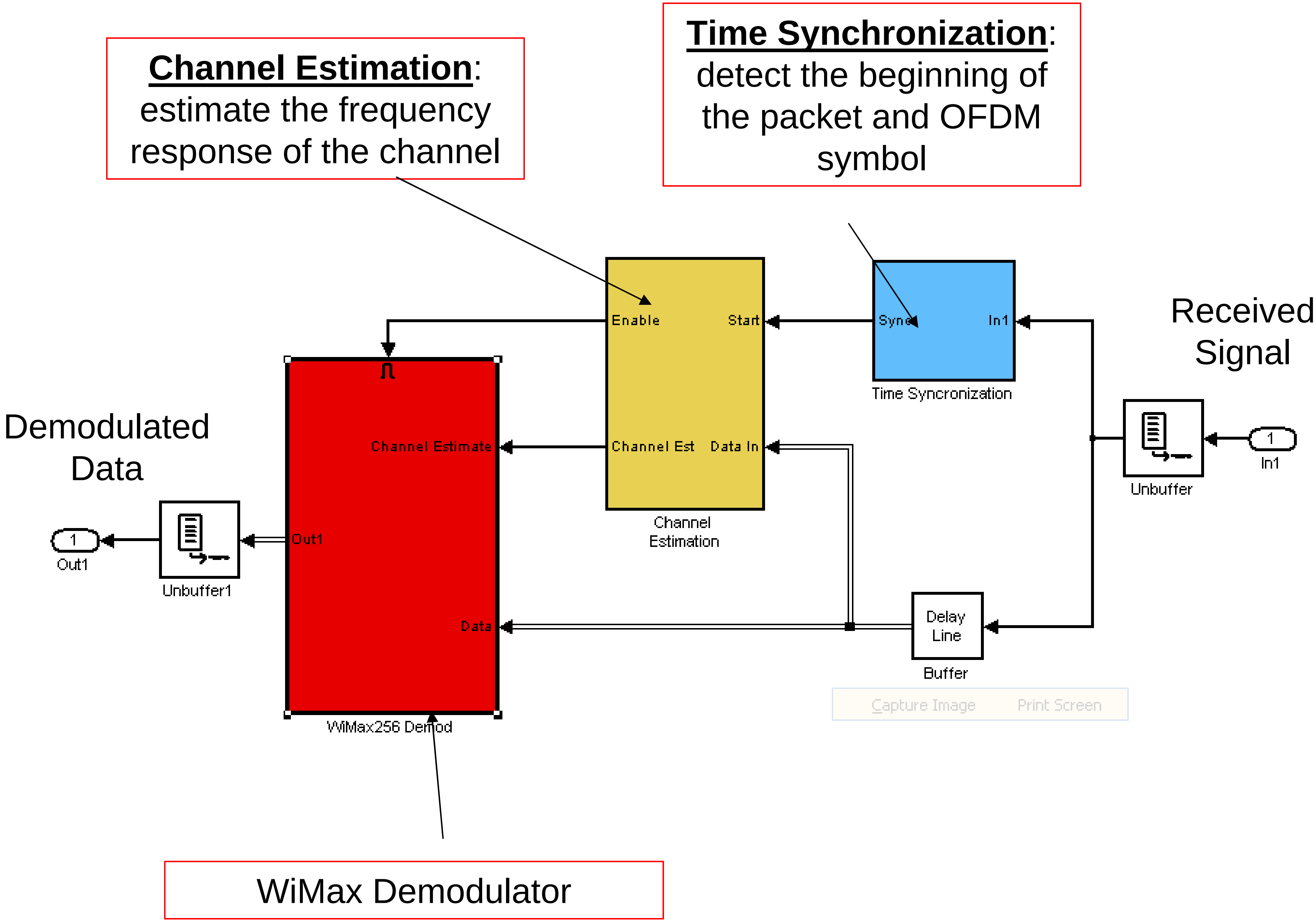
In addition to OFDM Modulator/Demodulator and Coding we need

- Time Synchronization: to detect when the packet begins
- Channel Estimation: needed in OFDM demodulator
- Channel Tracking: to track the time varying channel (for mobile only)

In addition we need

- Frequency Offset Estimation: to compensate for phase errors and noise in the oscillators
- Offset tracking: to track synchronization errors

Basic Structure of the Receiver

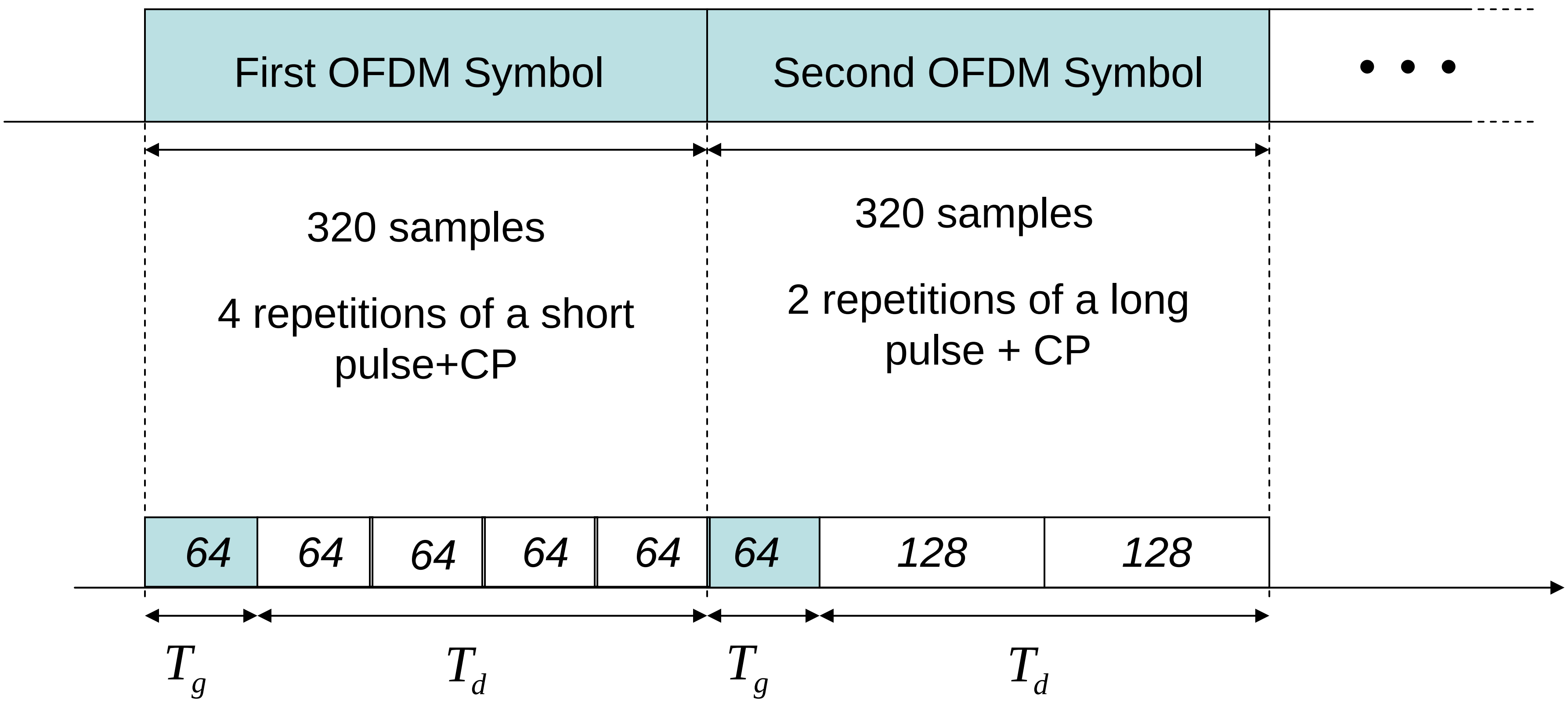


Time Synchronization

In IEEE802.16 (256 carriers, 64 CP) Time and Frequency Synchronization are performed by the Preamble.

Long Preamble: composed of 2 OFDM Symbols

Short Preamble: only the Second OFDM Symbol

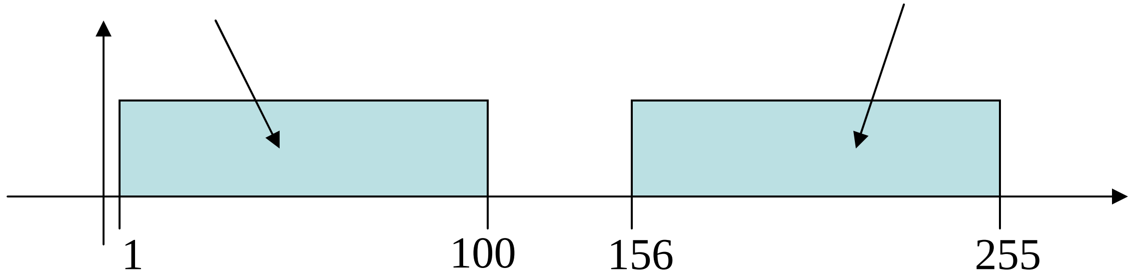


The standard specifies the Down Link preamble as QPSK for subcarriers between -100 and +100:

$$P_{ALL}[k] = \begin{cases} \pm 1 \pm j, & k = -100,\ldots,-1,+1,\ldots,+100 \\ 0, & \text{otherwise} \end{cases}$$

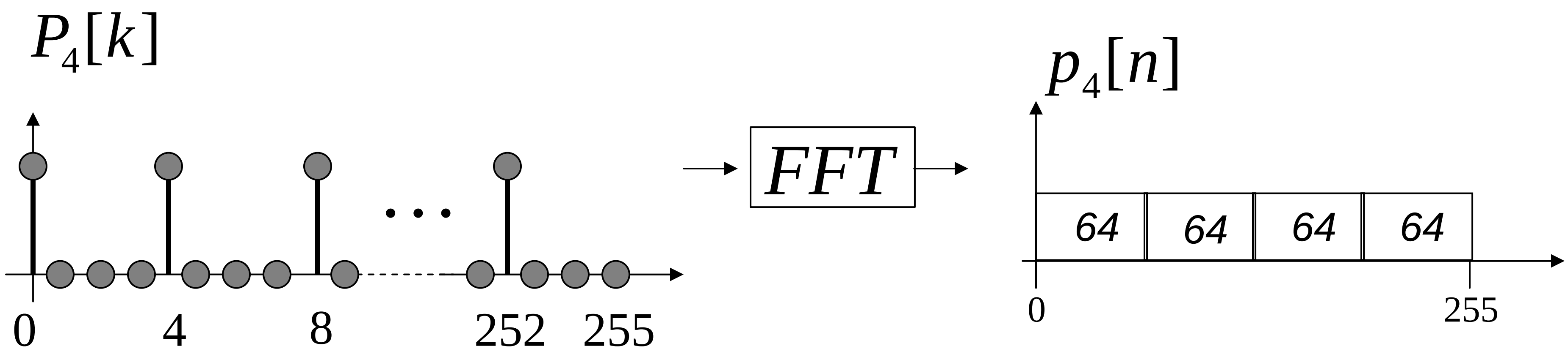
Using the periodicity of the FFT:

$$P_{ALL}[k],\ k = 1,\ldots,100 \qquad P_{ALL}[k] = P_{ALL}[256+k],\ k = -100,\ldots,-1$$

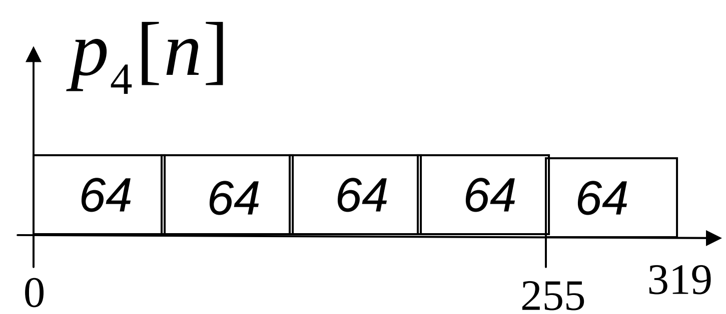


• **Short Preamble**, to obtain the 4 repetitions, choose only subcarriers multiple of 4:

$$P_4[k] = \begin{cases} 2P_{ALL}^*[k], & \text{if } k \bmod 4 = 0 \\ 0, & \text{otherwise} \end{cases}$$

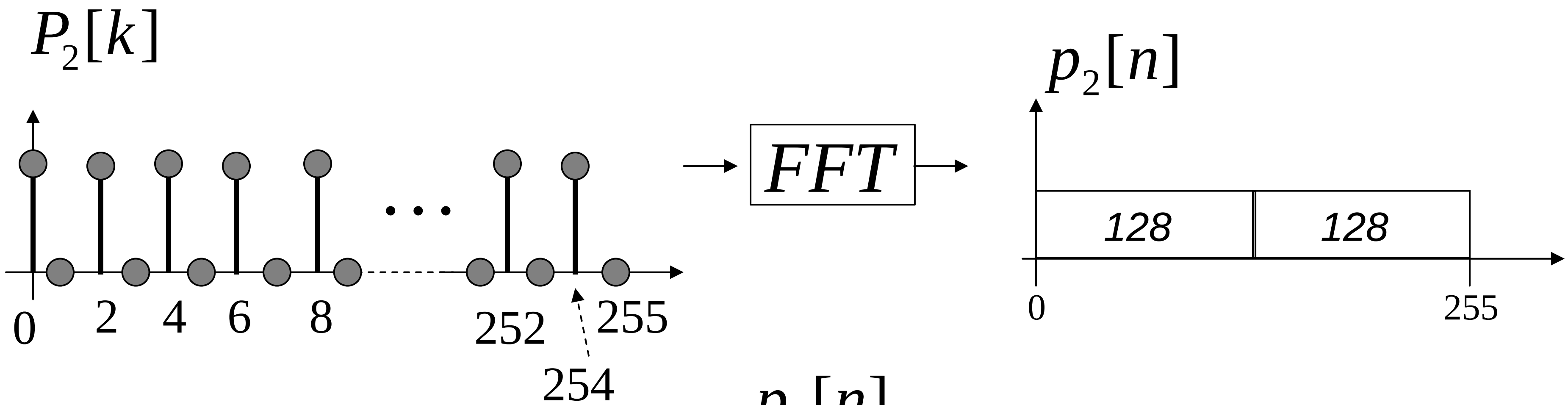


Add Cyclic Prefix:

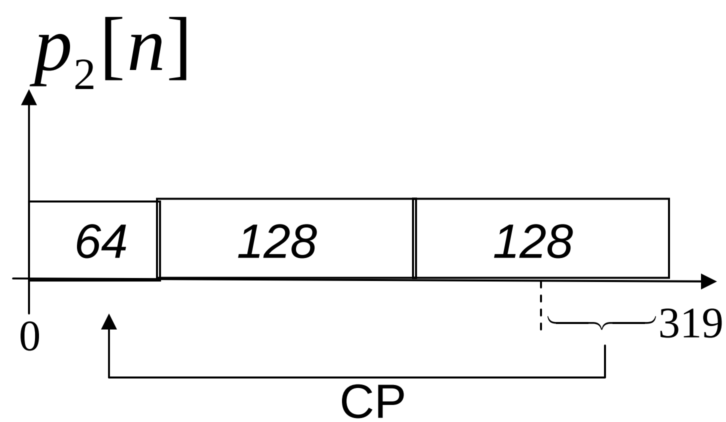


• **Long Preamble**: to obtain the 2 repetitions, choose only subcarriers multiple of 2 :

$$P_2[k]=\begin{cases}\sqrt{2}P_{ALL}[k], & \text{if } k \bmod 2 = 0\\ 0, & \text{otherwise}\end{cases}$$



Add Cyclic Prefix:



Several combinations for Up Link, Down Link and **Multiple Antennas**.

We can generate a number of preambles as follows:

With 2 Transmitting Antennas:

$$P_2^0[k] = \begin{cases} \sqrt{2} P_{ALL}[k], & \text{if } k \bmod 2 = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$P_2^1[k] = \begin{cases} \sqrt{2} P_{ALL}[k], & \text{if } k \bmod 2 = 1 \\ 0, & \text{otherwise} \end{cases}$$

With 4 Transmitting Antennas:

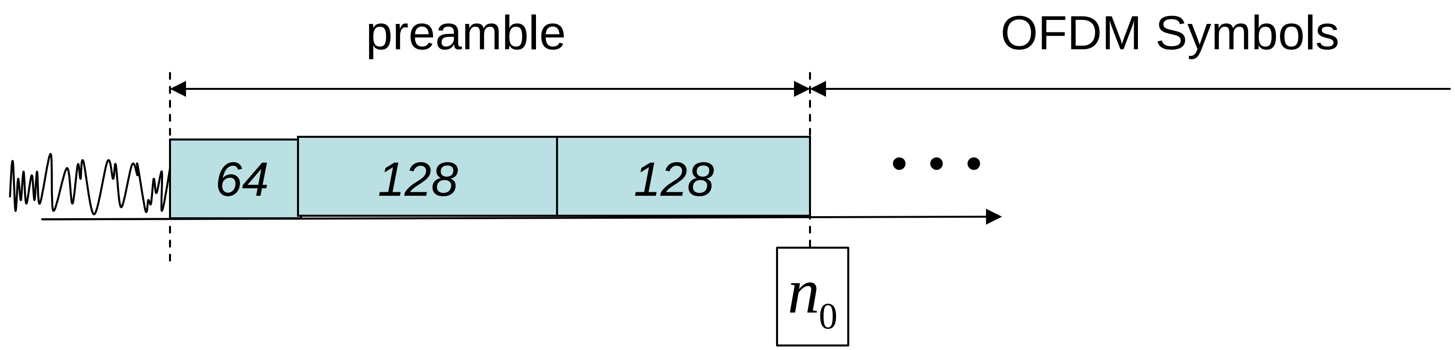
$$m = 0 \qquad P_4^0[k] = \begin{cases} 2 P_{ALL}^*[k], & \text{if } k \bmod 4 = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$m = 1,2,3 \qquad P_4^m[k] = \begin{cases} P_{ALL}^*[k+2-m], & \text{if } k \bmod 4 = m \\ 0, & \text{otherwise} \end{cases}$$

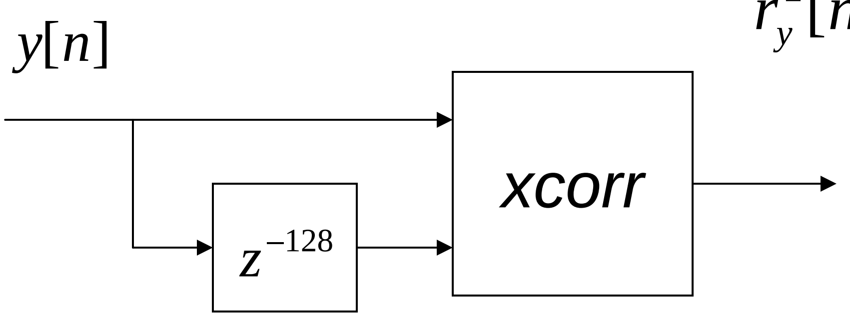
Time Synchronization from Long Preamble

1. Coarse Time Synchronization using Signal Autocorrelation

Received signal:



Compute Crosscorrelation Coefficient:

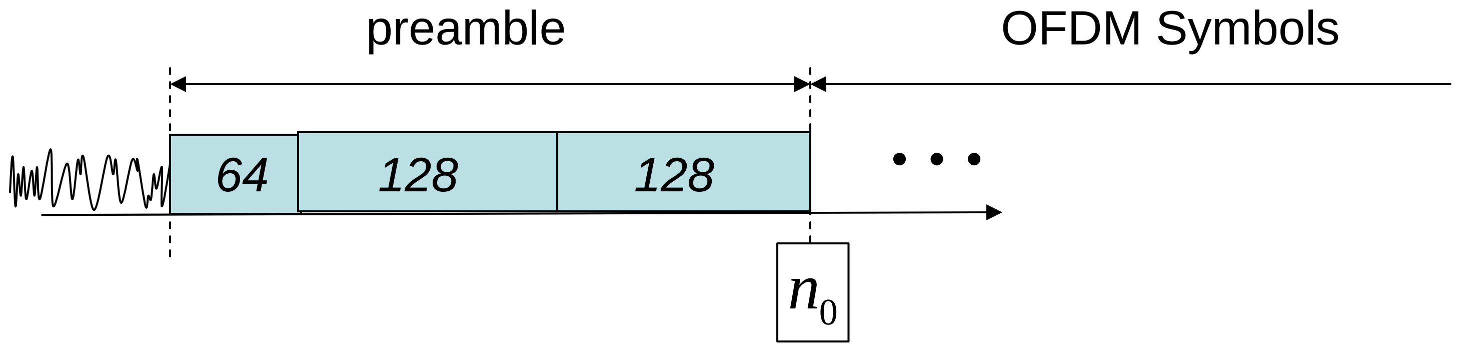


$$r_y^2[n] = \frac{\left| \sum_{\ell=0}^{127} y[n-\ell]\, y^*[n-128-\ell] \right|^2}{\left(\sum_{\ell=0}^{127} \left| y[n-\ell] \right|^2 \right) \times \left(\sum_{\ell=0}^{127} \left| y[n-128-\ell] \right|^2 \right)}$$

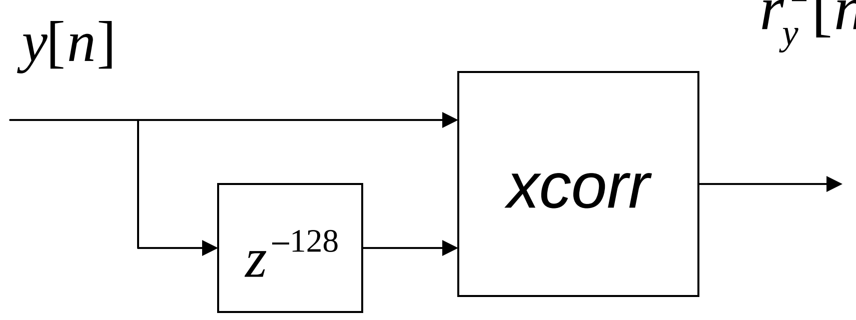
Time Synchronization from Long Preamble

1. Coarse Time Synchronization using Signal Autocorrelation

Received signal:

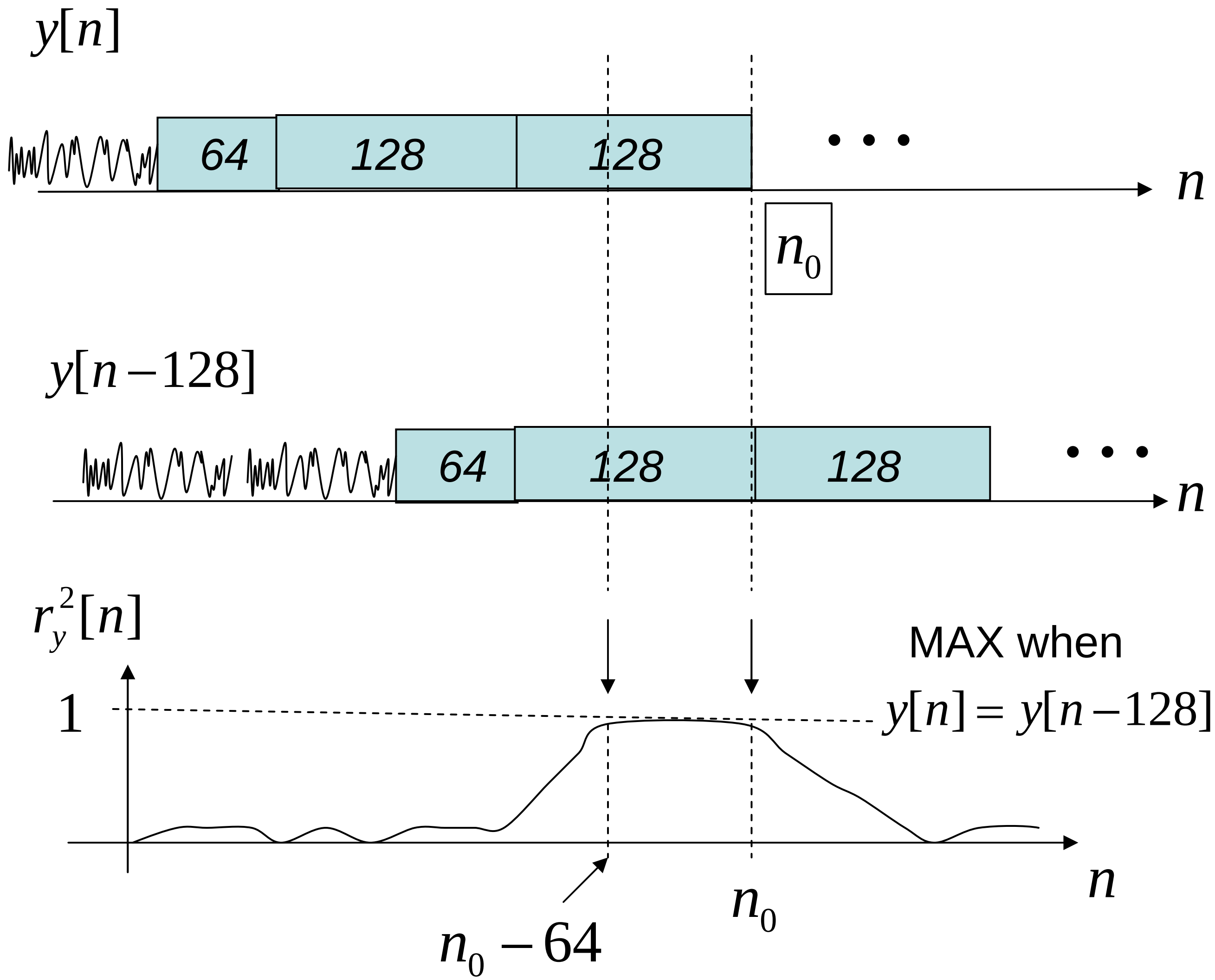


Compute Autocorrelation :



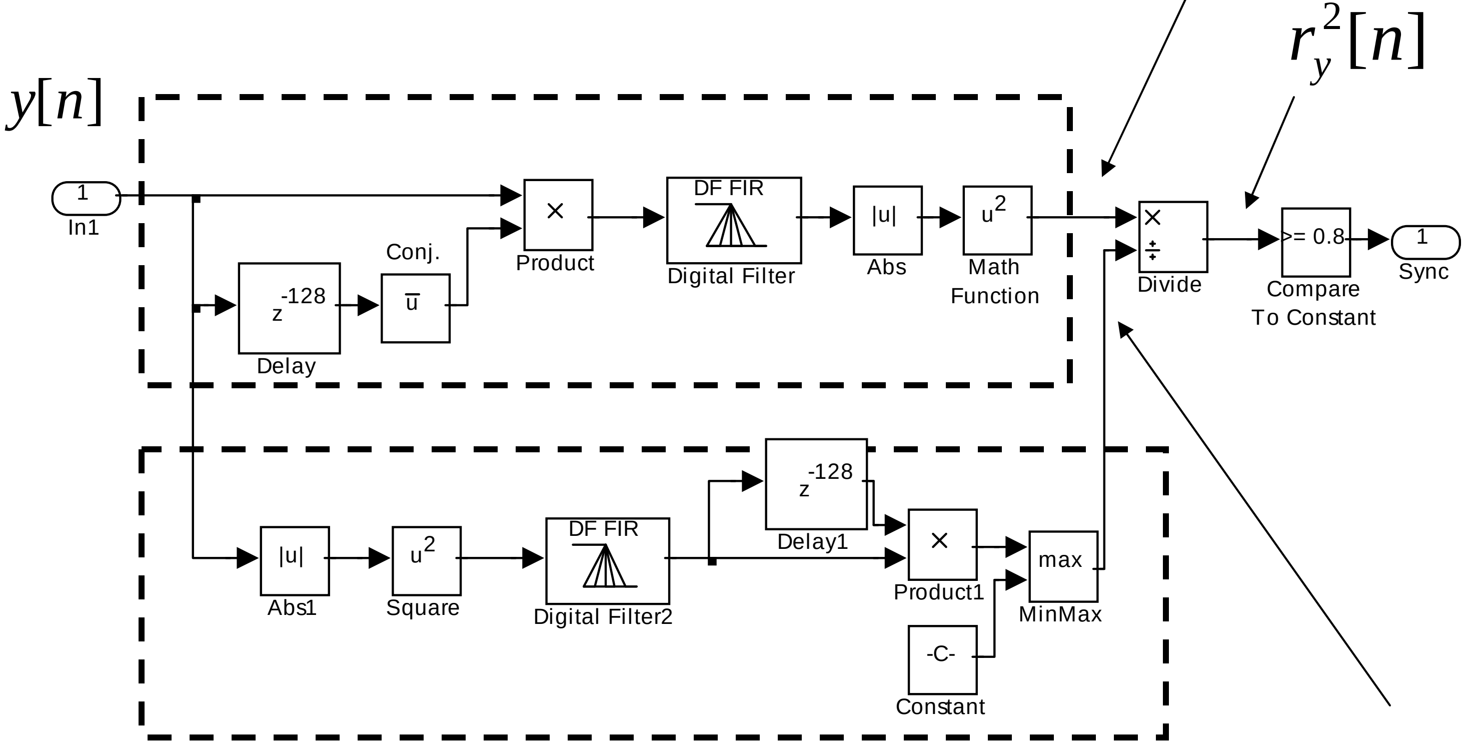
$$r_y^2[n] = \frac{\left|\sum_{\ell=0}^{127} y[n-\ell]\,y^*[n-128-\ell]\right|^2}{\left(\sum_{\ell=0}^{127} \left|y[n-\ell]\right|^2\right) \times \left(\sum_{\ell=0}^{127} \left|y[n-128-\ell]\right|^2\right)}$$

Effect of Periodicity on Autocorrelation:



Compute it in Simulink:

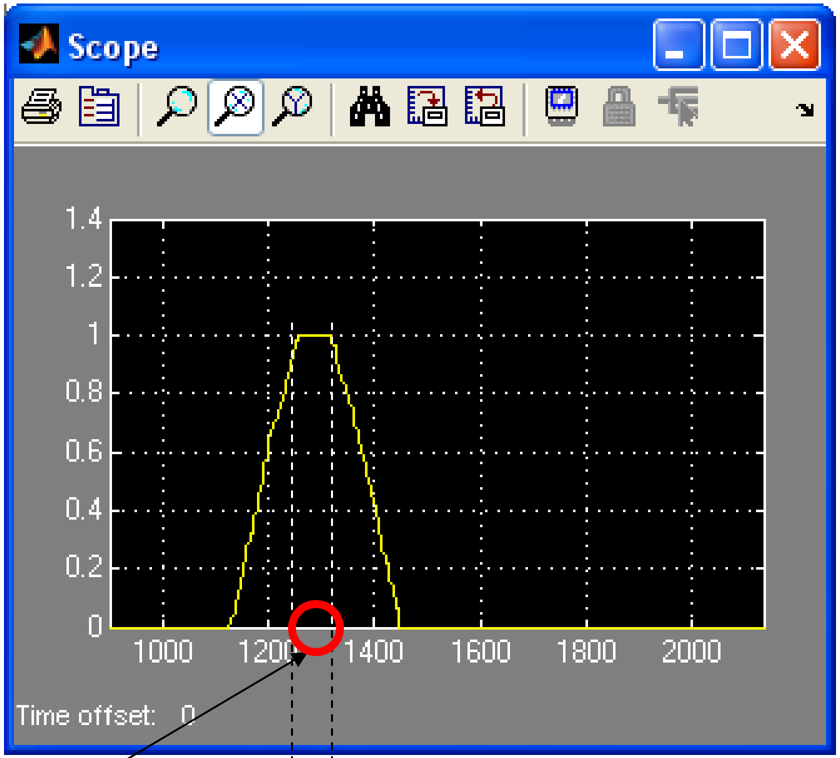
$$\left|\sum_{\ell=0}^{127} y[n-\ell]\,y^*[n-128-\ell]\right|^2$$



$$\left(\sum_{\ell=0}^{127} \left|y[n-\ell]\right|^2\right)\left(\sum_{\ell=0}^{127} \left|y[n-128-\ell]\right|^2\right)$$

Example: perfect channel (no spread in time)

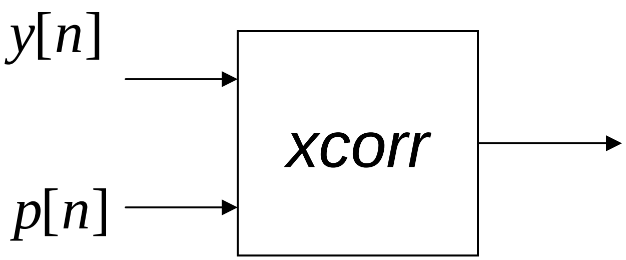
$r_y[n]$



64

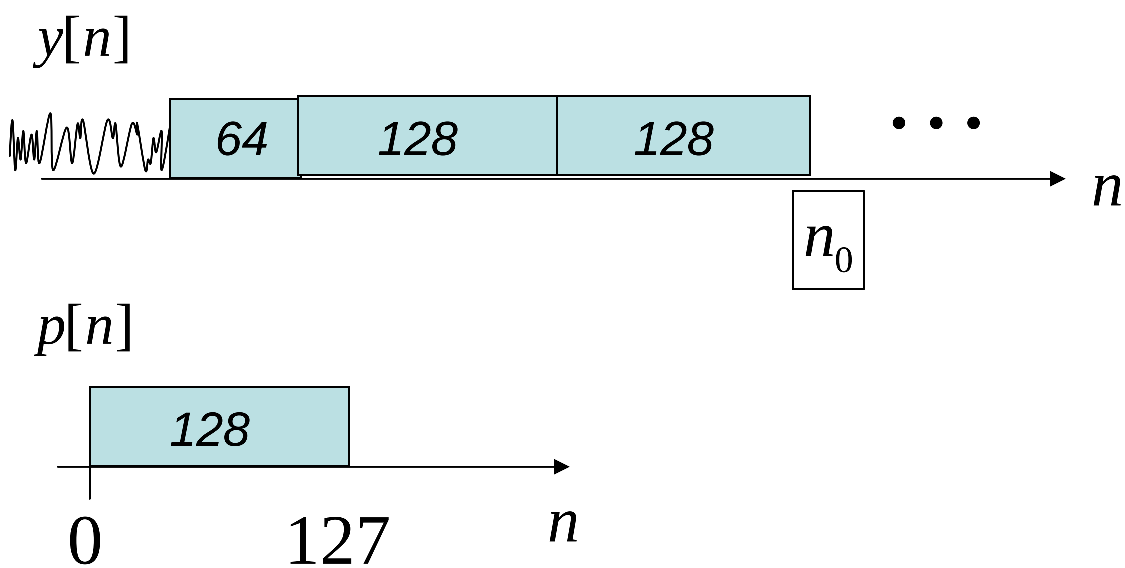
The packet begins
somewhere in here

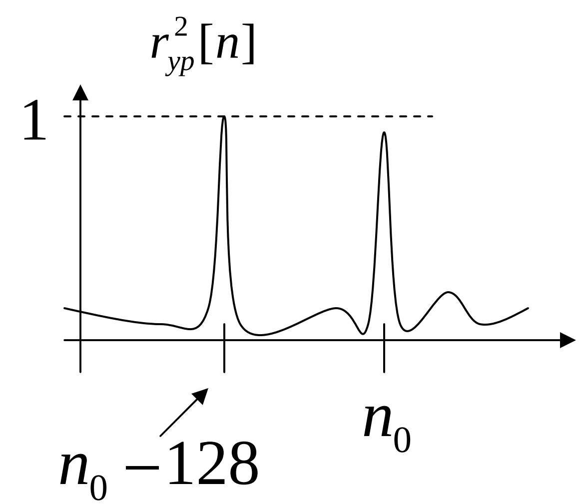
2. Fine Time Synchronization using Cross Corelation with Preamble



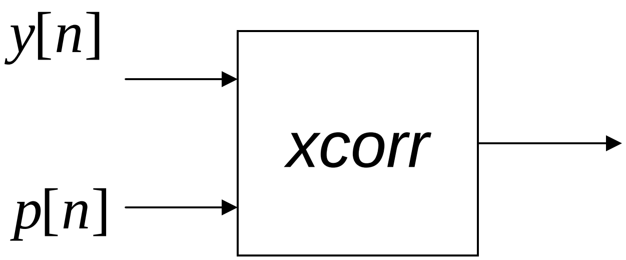
$$r_{yp}[n] = \frac{\left| \sum_{l=0}^{127} y[n-\ell] p^*[\ell] \right|^2}{\left(\sum_{l=0}^{127} \left| y[n-\ell] \right|^2 \right) \left(\sum_{l=0}^{127} \left| p[\ell] \right|^2 \right)}$$

Since the preamble is random (almost like white noise), it has a short autocorrelation:



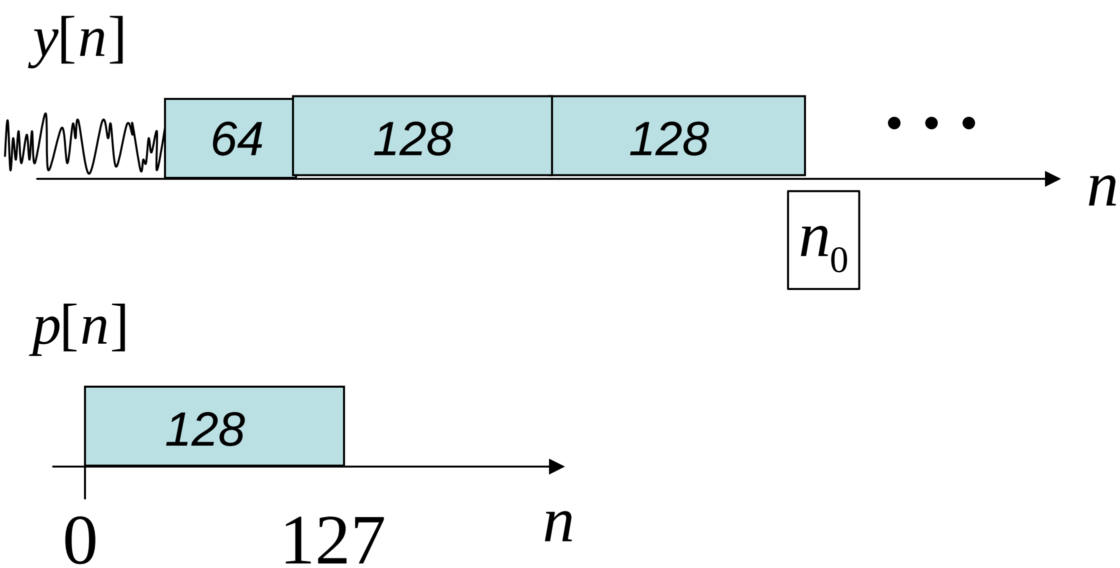


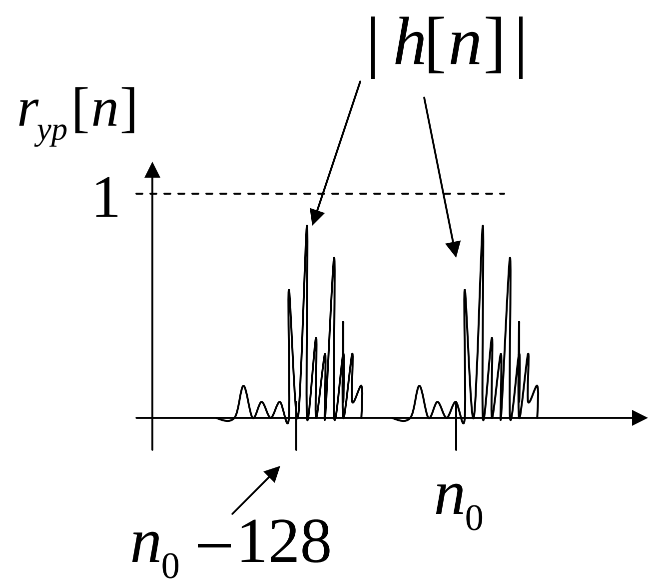
… with dispersive channel



$$r_{yp}[n] = \frac{\left|\sum_{l=0}^{127} y[n-\ell]p^*[\ell]\right|^2}{\left(\sum_{l=0}^{127} |y[n-\ell]|^2\right)\left(\sum_{l=0}^{127} |p[\ell]|^2\right)}$$

Since the preamble is random, almost white, recall that the crosscorrelation yields the impulse response of the channel





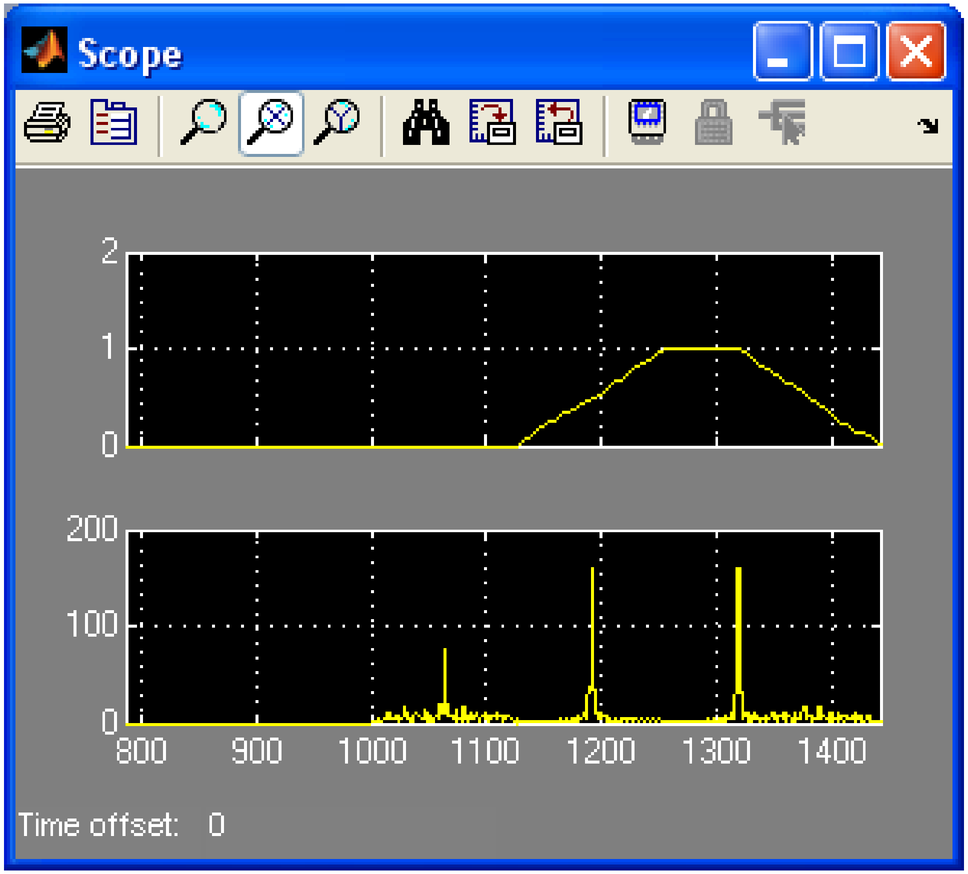
Compare the two (non dispersive channel):

Autocorrelation of received data

r_y

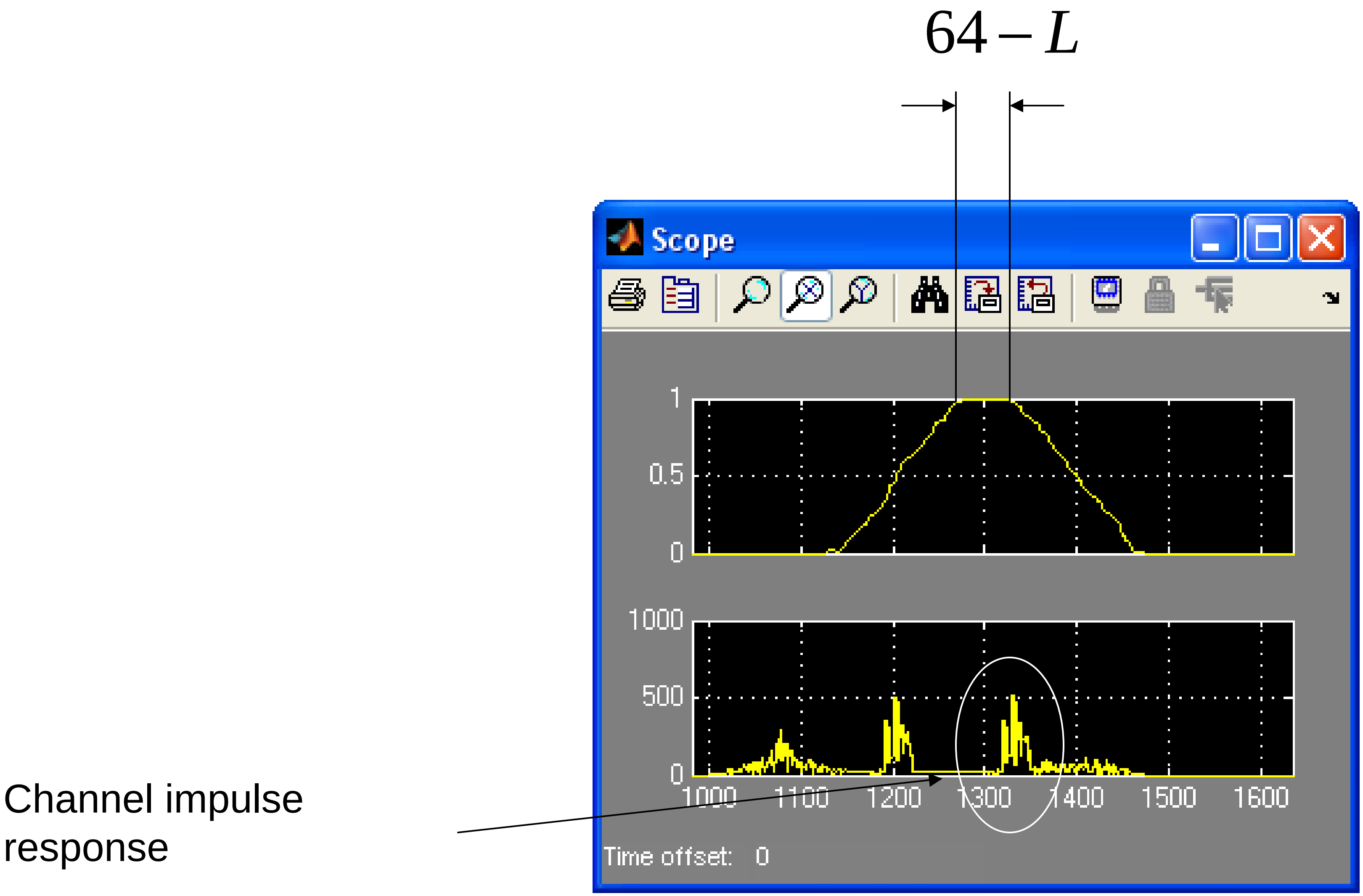
Crosscorrelation with preamble

r_{yp}

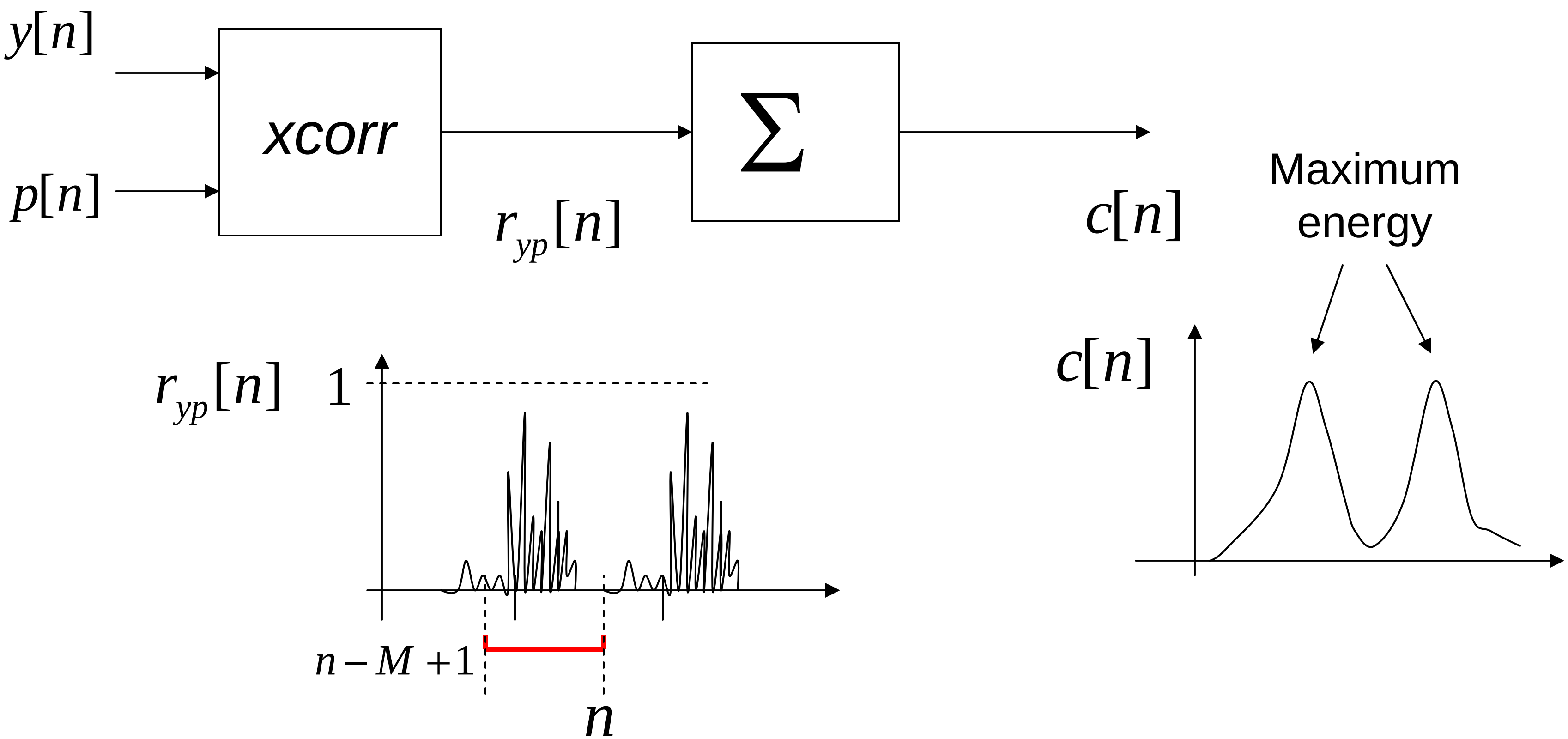


Synchronization with Dispersive Channel

Let L be the length of the channel impulse response



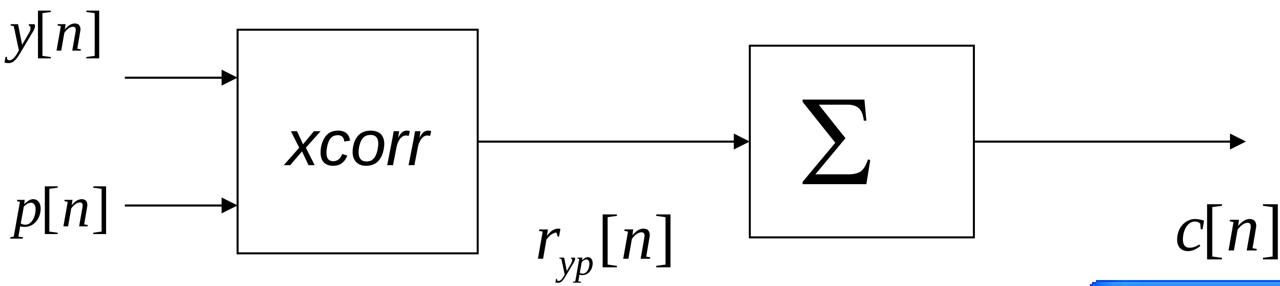
In order to determine the starting point, compute the energy on a sliding window and choose the point of maximum energy

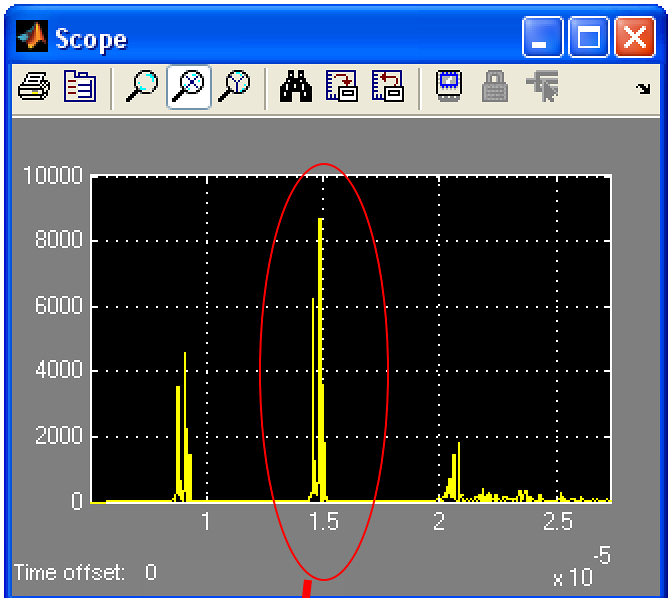


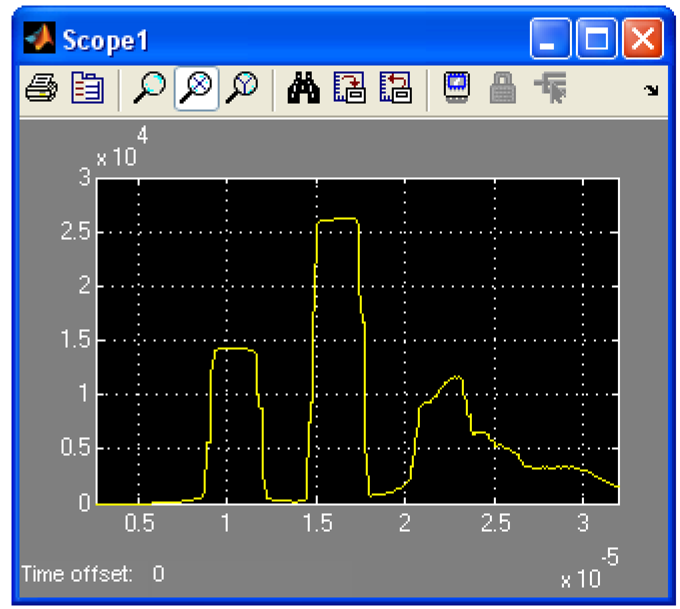
M=max length of channel = length of CP

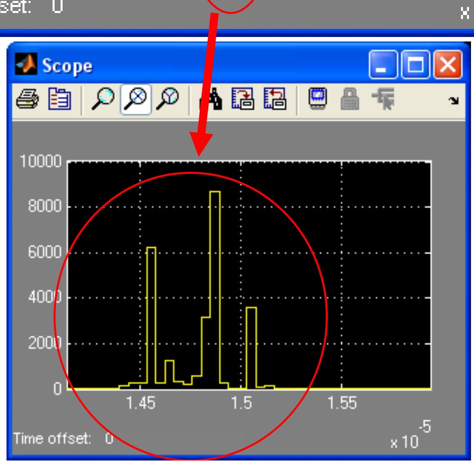
$$c[n] = \sum_{k=0}^{M-1} r_{yp}[n-k]$$

Example

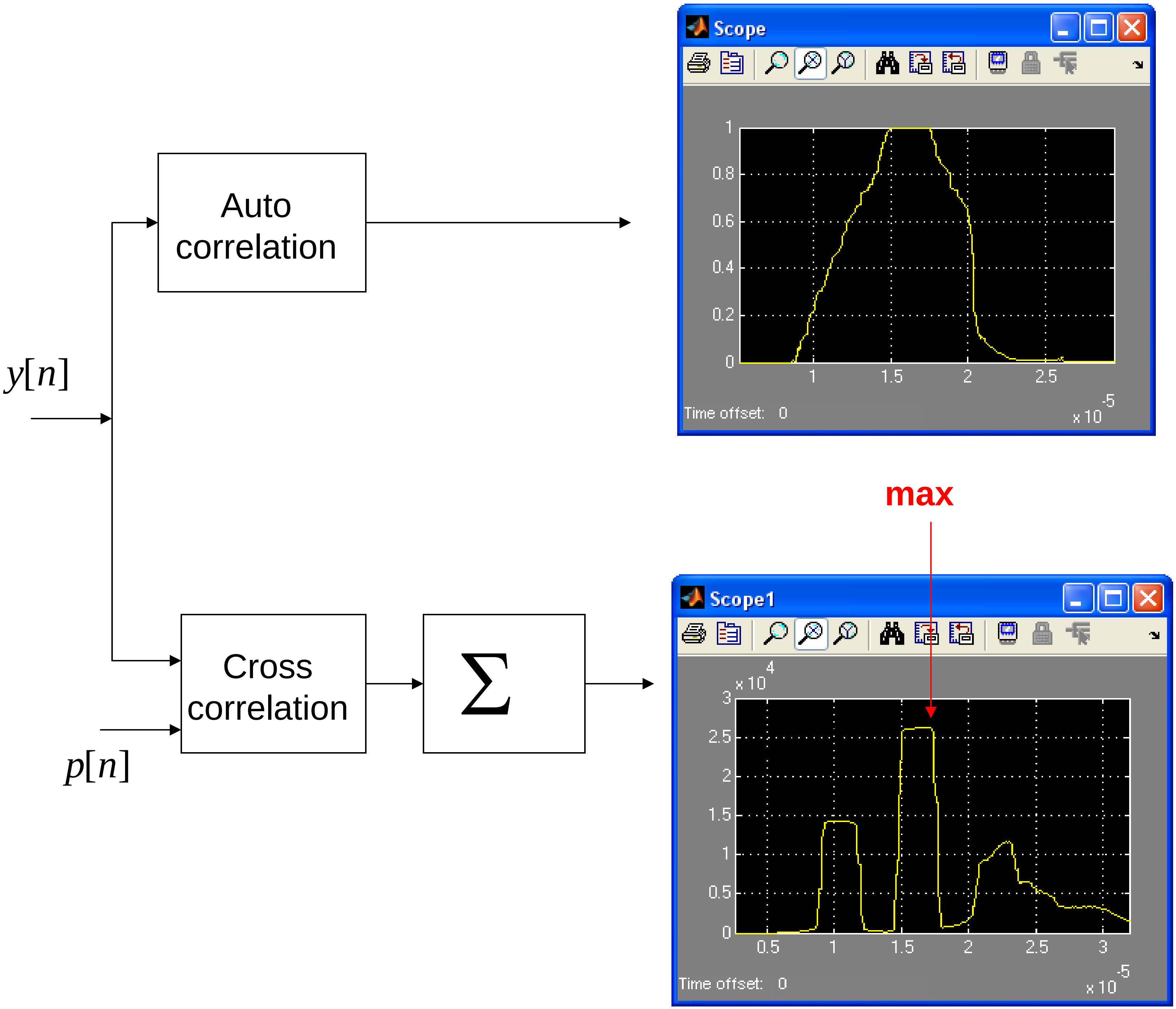




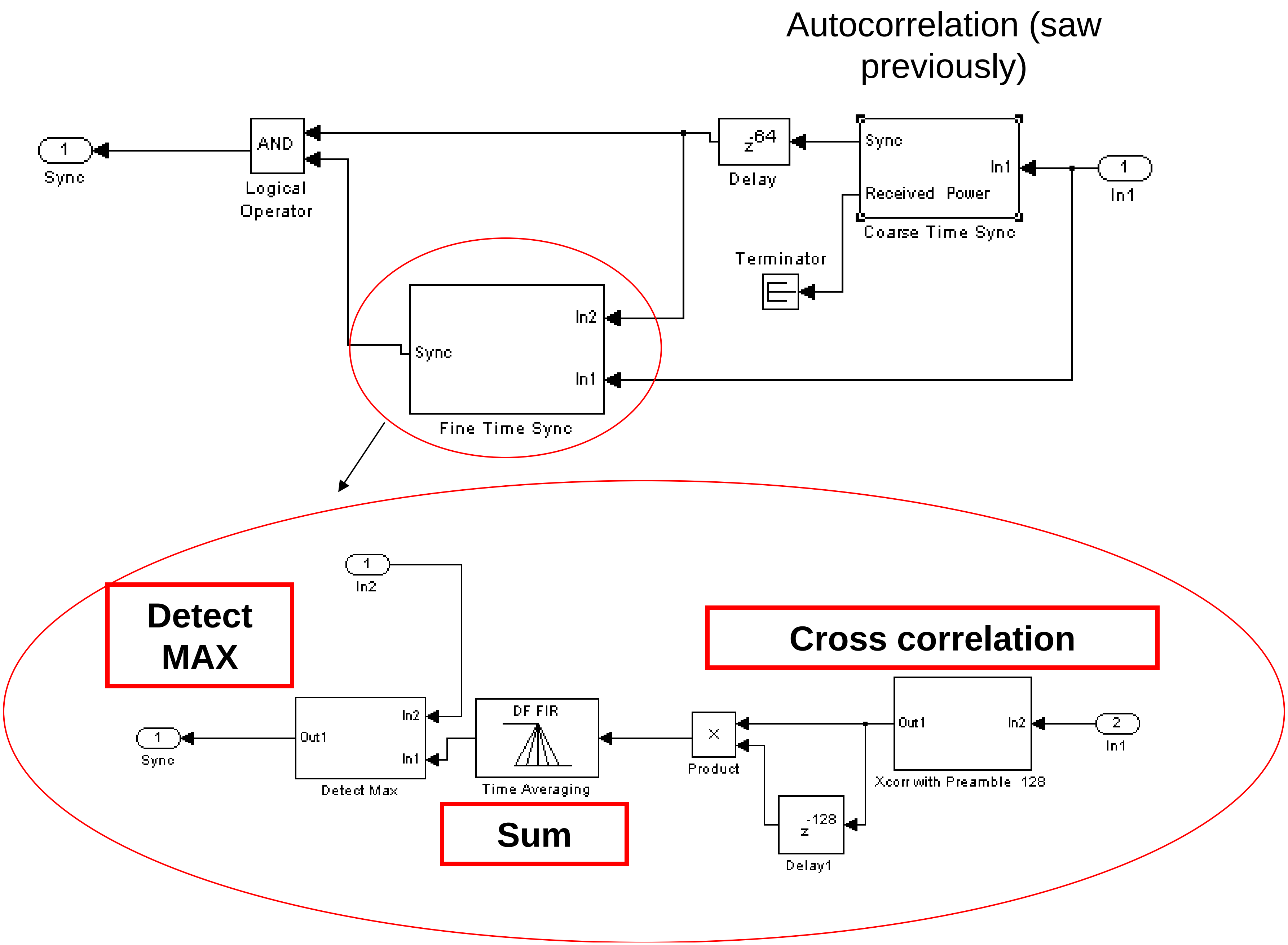




Impulse response
of channel

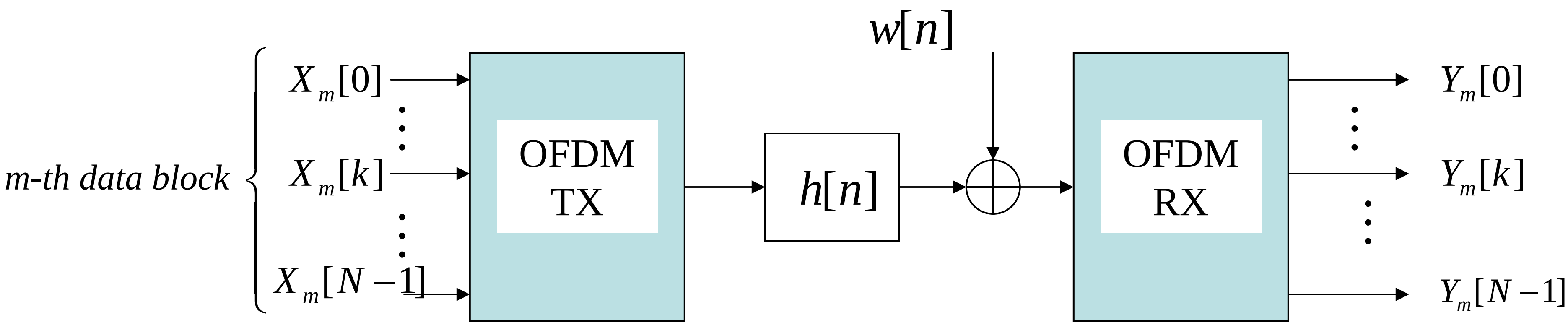


Auto correlation
y[n]
Cross correlation
p[n]
Σ
Scope
Time offset: 0
Scope1
Time offset: 0
max


Autocorrelation (saw previously)
1
Sync
AND
Logical Operator
z^{-64}
Delay
Sync
In1
Received Power
Coarse Time Sync
1
In1
Terminator
In2
Sync
In1
Fine Time Sync
Detect MAX
1
In2
Cross correlation
1
Sync
Out1
In2
In1
Detect Max
DF FIR
Time Averaging
Product
Out1
In2
Xcorr with Preamble 128
2
In1
z^{-128}
Delay1
Sum

Channel Estimation

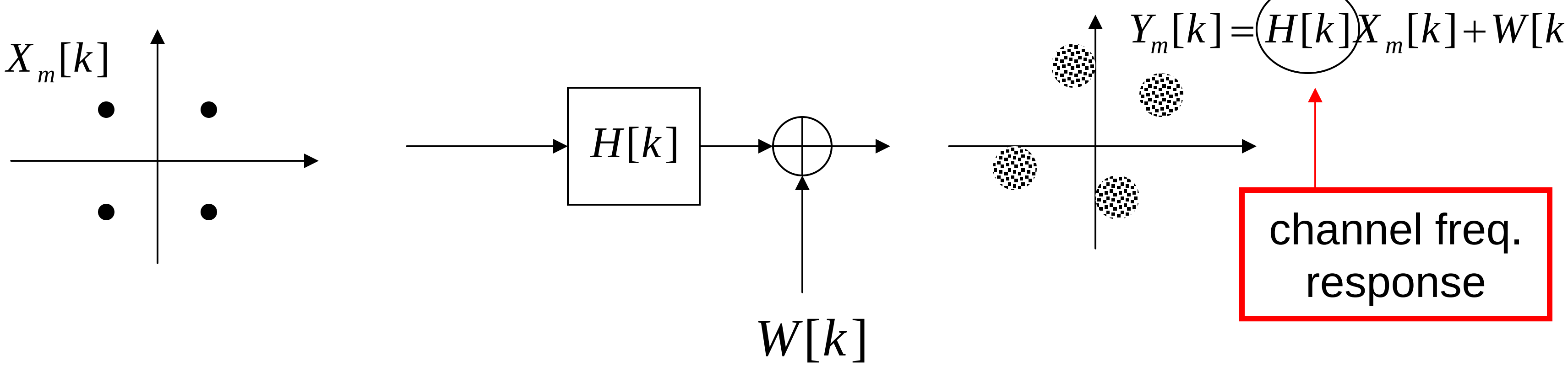
Recall that, at the receiver, we need the frequency response of the channel:



Transmitted:

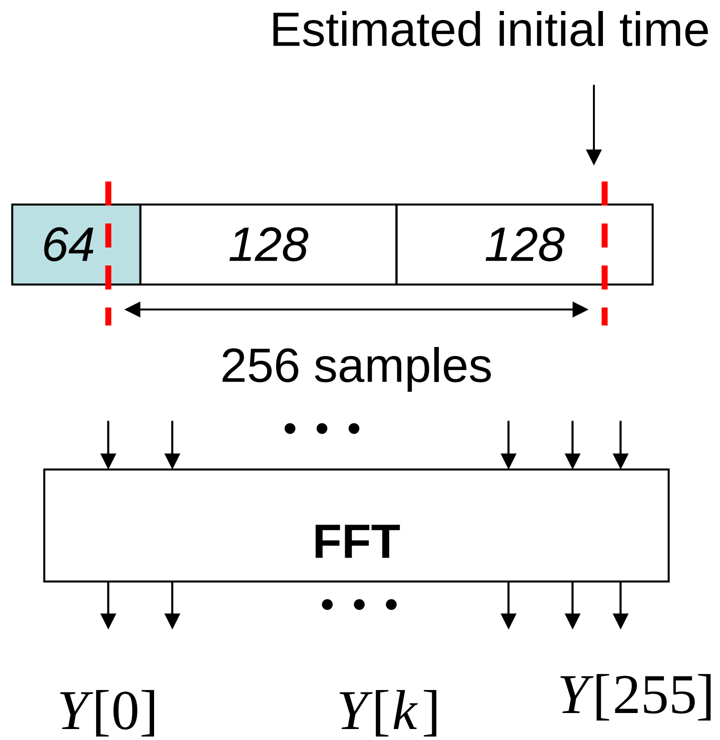
Received:

$$Y_m[k] = H[k] X_m[k] + W[k]$$



From the Preamble: at the beginning of the received packet. The transmitted signal in the preamble is known at the receiver: after time synchronization, we take the FFT of the received preamble

Received Preamble:



$$Y[k] = H[k]X_p[k] + W[k],\ k = 0,\ldots,255$$

$$Y[k] = H[k]X_p[k] + W[k],\; k = 0,\ldots,255$$

Solve for $H[k]$ using a Wiener Filter (due to noise):

$$\hat{H}[k] = \frac{Y[k]X^{*}_{preamble}[k]}{|X_p[k]|^2 + \sigma_w^2}$$

noise covariance

Problem: when $X_p[k] = 0$ we cannot compute the corresponding frequency response $H[k]$

Fact: by definition,

$X_p[k] = \pm 1 \pm j$	**if**	$k = 2,4,\ldots,100$ $k = 156,158,\ldots,254$
$X_p[k] = 0$		**otherwise (ie DC, odd values, frequency guards)**

Two solutions:

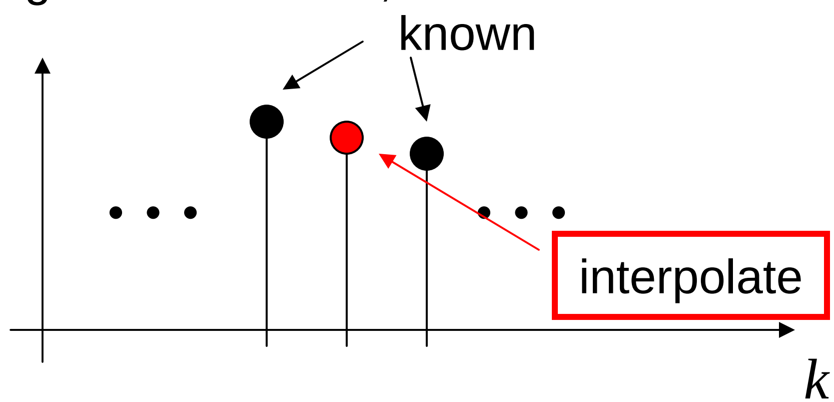
1. Compute the channel estimate

$$\hat{H}[k] = \frac{Y[k] X^*_{preamble}[k]}{|X_p[k]|^2 + \sigma_w^2}$$

only for the frequencies *k* such that

$$X_p[k] \neq 0$$

and interpolate for the other frequencies. This might not yield good results and the channel estimate might be unreliable;



2. Recall the FFT and use the fact that we know the maximum length *L* of the channel impulse response

$$Y[k] = H[k]X_p[k] + W[k]$$

Since the the preamble is such that either $|X_p[k]| = 0$ or $|X_p[k]| = \sqrt{2}$

For the indices where $|X_p[k]| = \sqrt{2}$ we can write:

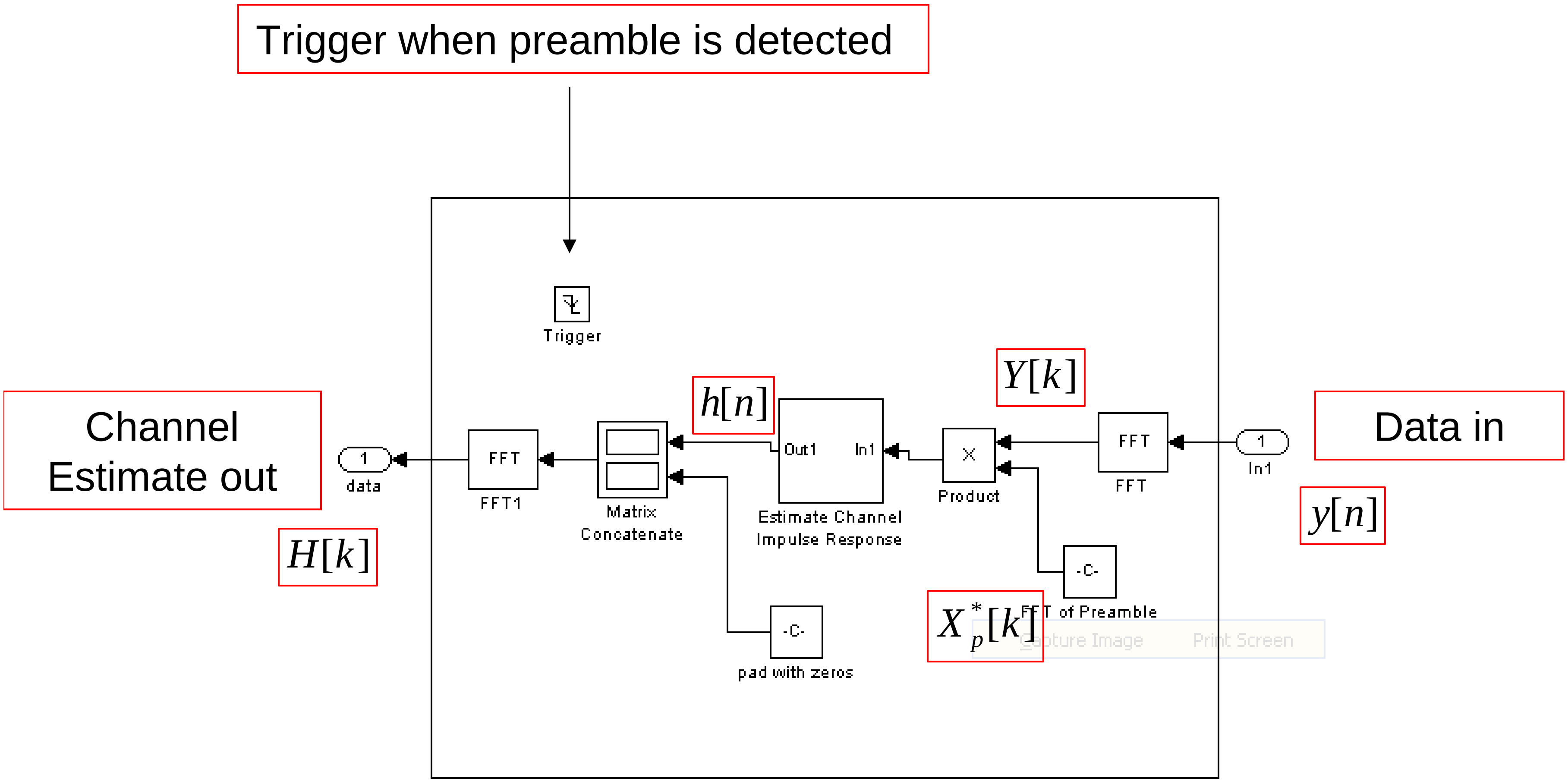
$$Y[k]X_p^*[k] = \left(\sum_{n=0}^{L-1} h[n] e^{-jk\frac{2\pi}{N}n} \right) \sqrt{2} + W[k]X_p^*[k] \quad \textbf{for} \quad \begin{array}{l} k = 2,4,\ldots,100 \\ k = 156,158,\ldots,254 \end{array}$$

so that we have **100 equations and *L*=64 unknowns.**

This is solved by a matrix multiplication

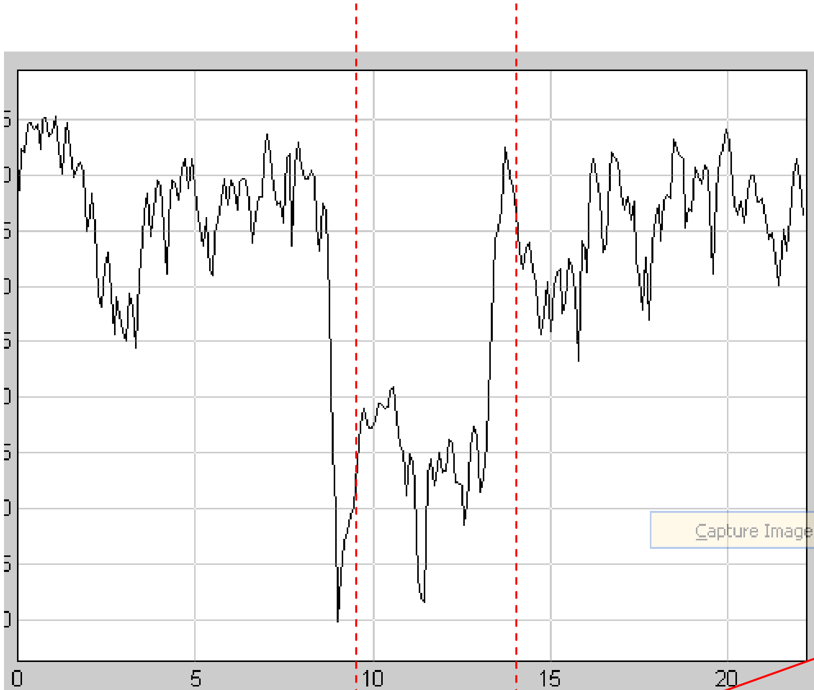
$$\boxed{\underset{64\times 1}{h} = F_{inv}\left(\underset{100\times 1}{YX_p^*}\right)}$$

Simulink Implementation



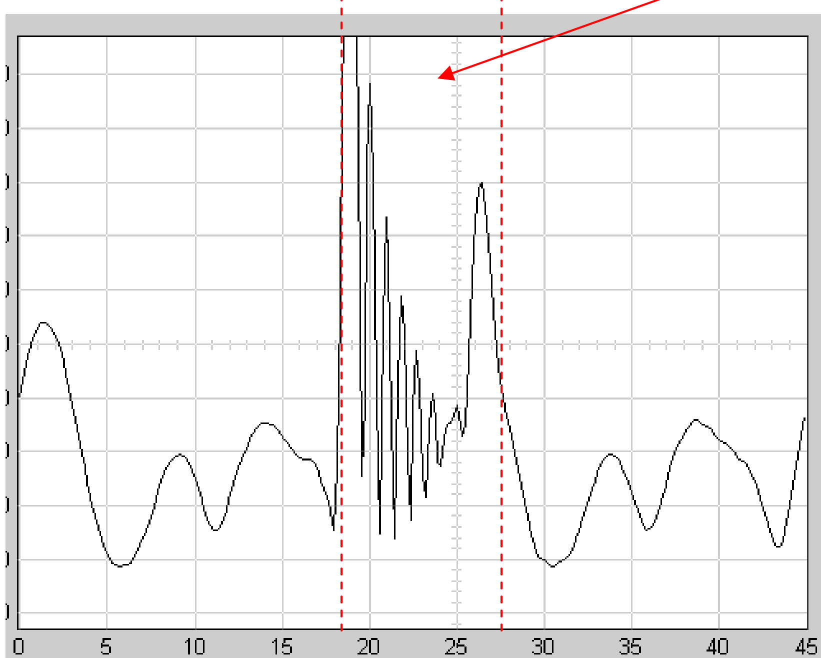
Example:

NOT TO SCALE



Spectrum of Received Signal

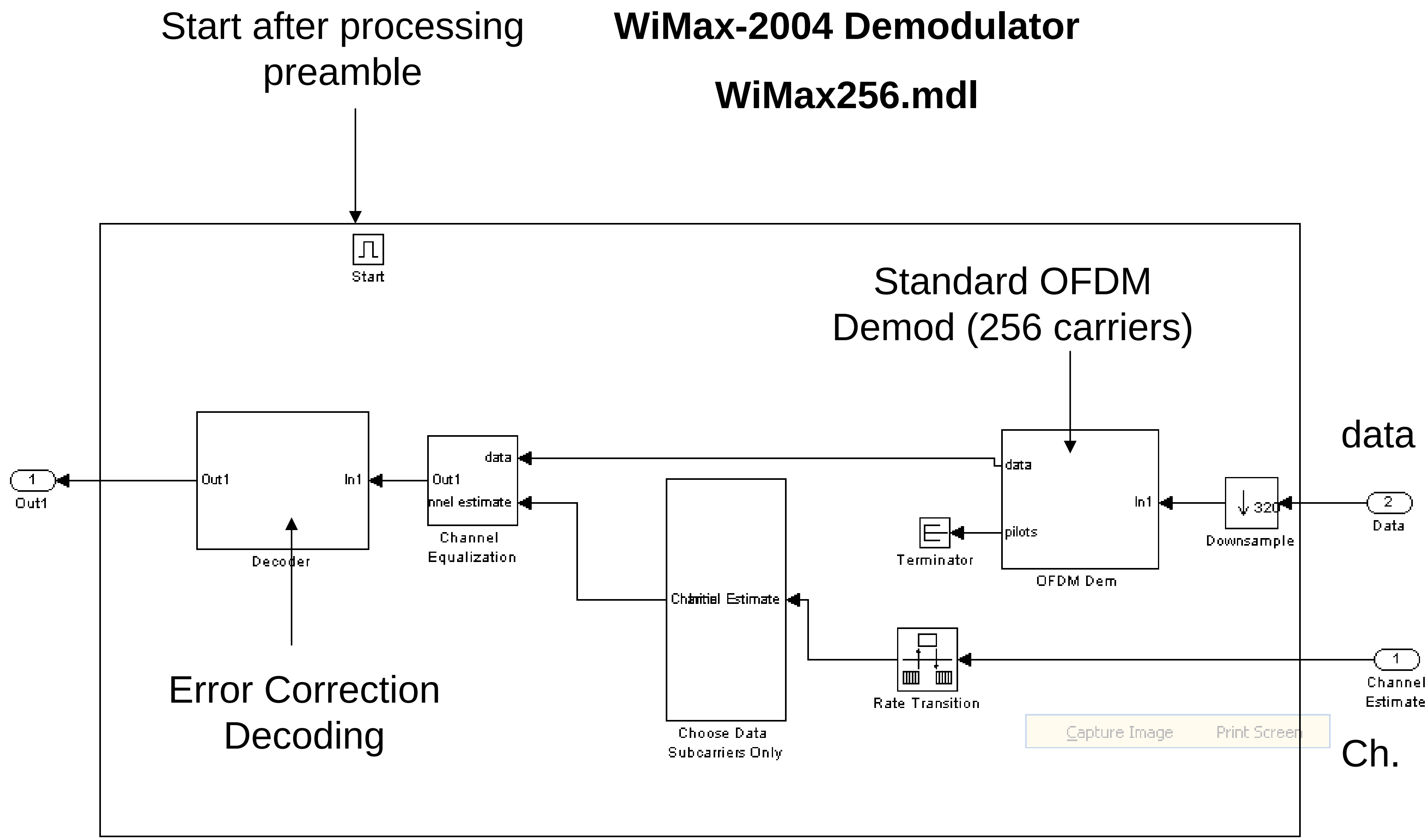
As expected, it does not match in the Frequency Guards



Estimated Frequency Response of Channel

WiMax-2004 Demodulator

WiMax256.mdl



Channel Tracking

In mobile applications, the channel changes and we need to track it.

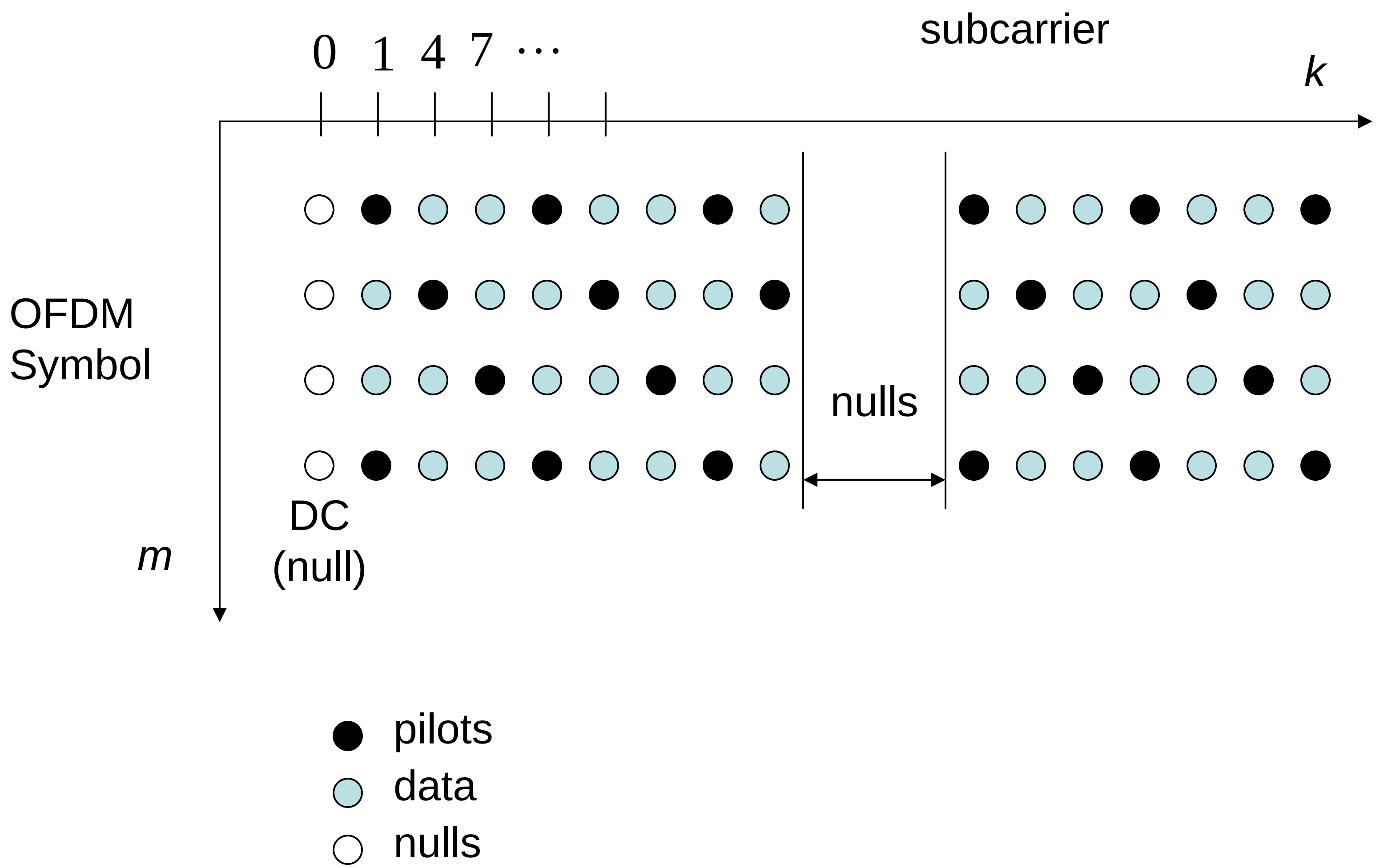
IEEE802.16-2005 tracks the channel by embedding pilots within the data.

In the FUSC (Full Use of Sub Carriers) scheme, the pilots subcarriers are chosen within the non-null subcarriers as

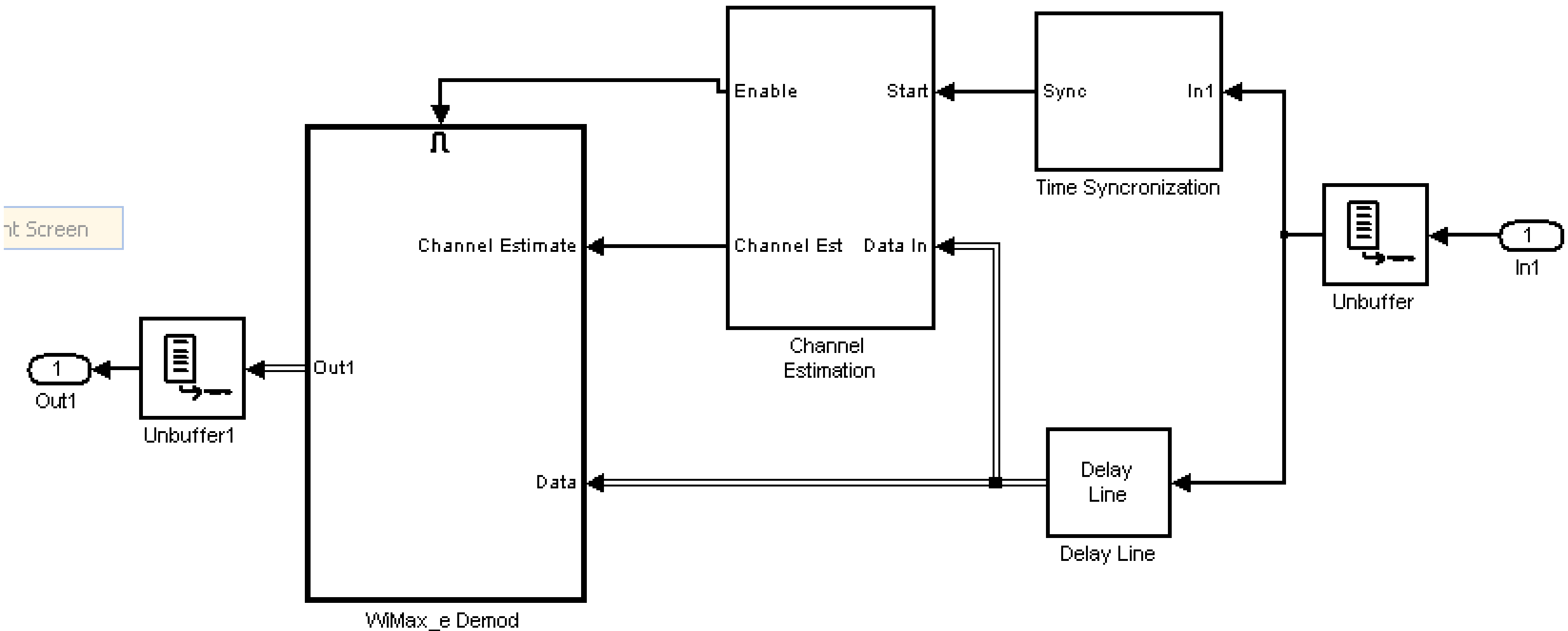
$$9k + 3m + 1$$

with $$m = [\mathrm{symbol\_index}] \bmod 3 = 0,1,2$$

$$k = \begin{cases} 0,\ldots,191 & \text{for } N_{FFT} = 2048 \\ 0,\ldots,95 & \text{for } N_{FFT} = 1024 \\ 0,\ldots,47 & \text{for } N_{FFT} = 512 \\ 0,\ldots,11 & \text{for } N_{FFT} = 128 \end{cases}$$

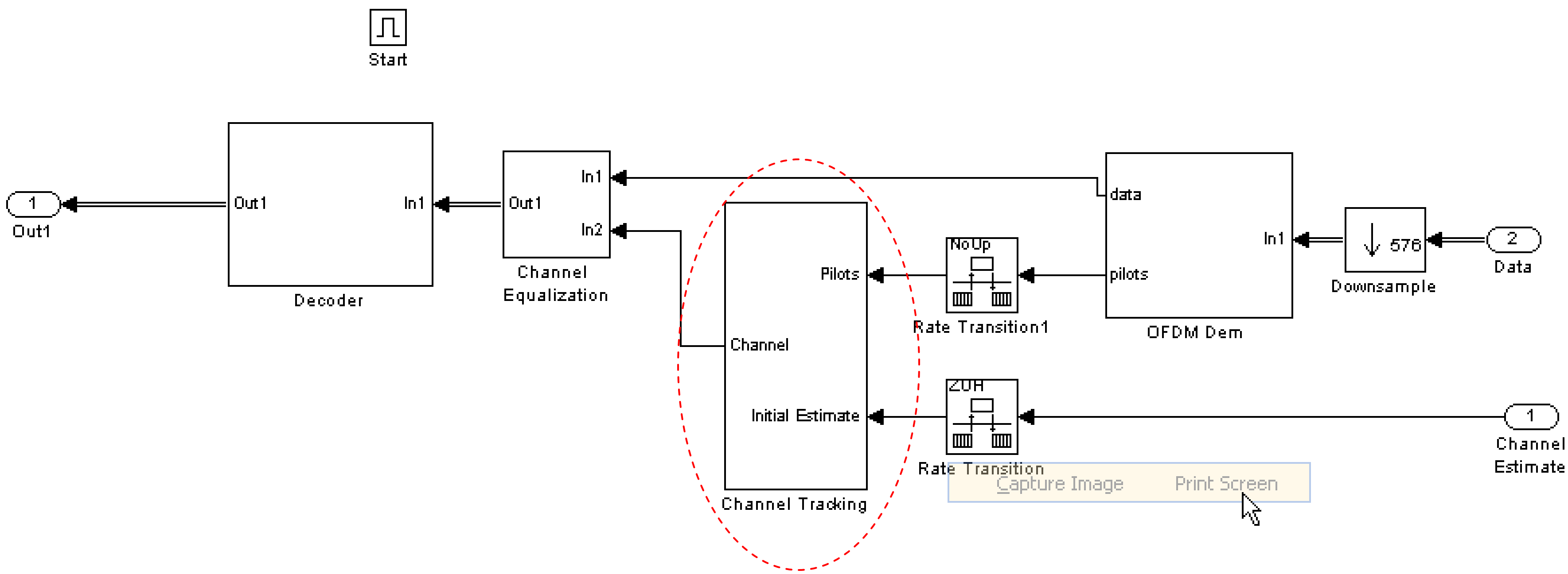


subcarrier
0 1 4 7 ···
k
OFDM Symbol
nulls
DC (null)
m
pilots
data
nulls



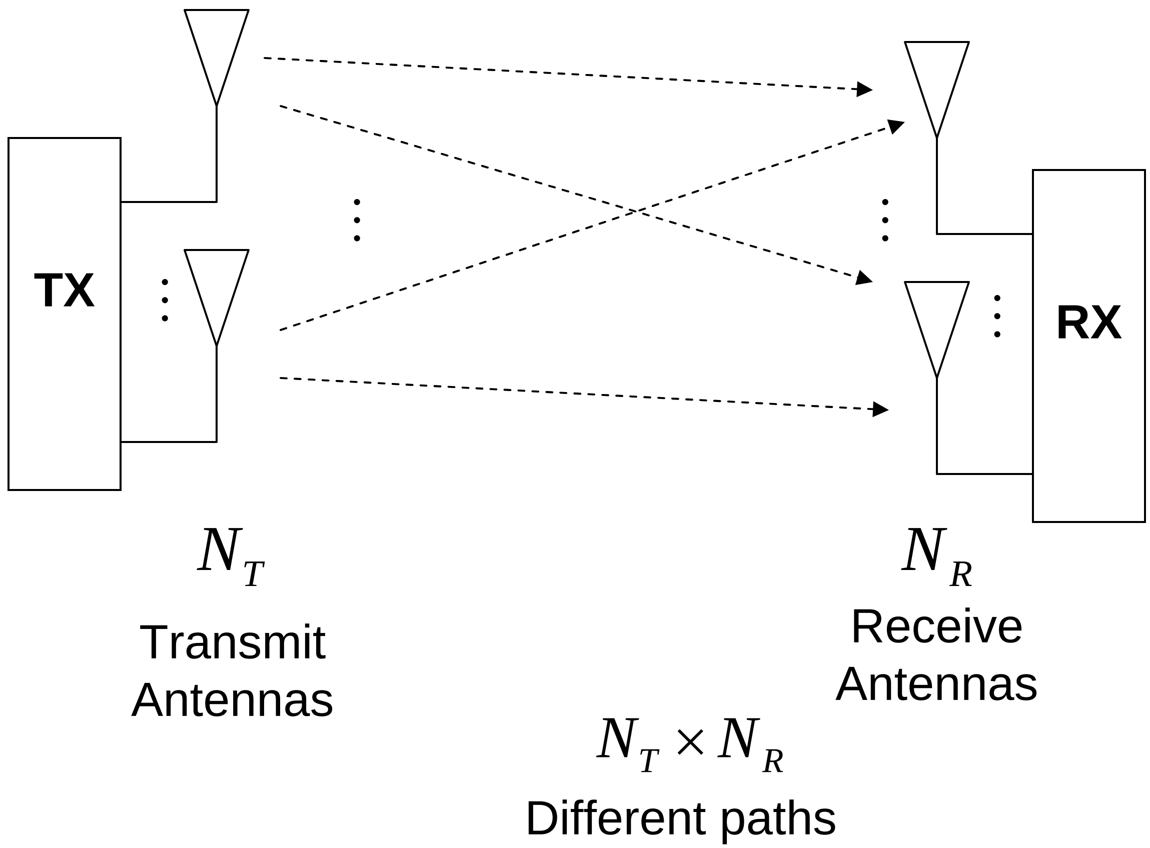
Expand the
Demodulator

Demodulator



Channel
tracking

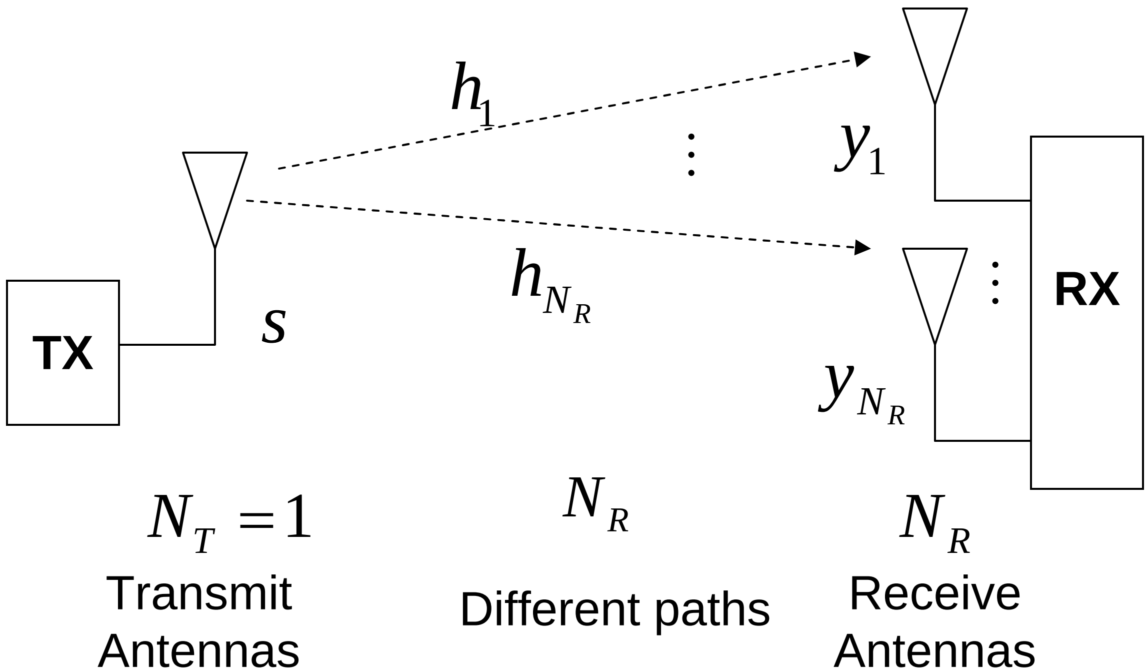
Multiple Antenna Systems (MIMO)



Two cases:

1. **Array Gain**: if all paths are strongly correlated to which other the SNR can be increased by array processing;
2. **Diversity Gain**: if all paths are uncorrelated, the effect of channel fading can be attenuated by diversity combining

Receive Diversity:



$$\begin{bmatrix} y_1 \\ \vdots \\ y_{N_R} \end{bmatrix} = \begin{bmatrix} h_1 \\ \vdots \\ h_{N_R} \end{bmatrix} \sqrt{E_S}\, s + \sqrt{N_0} \begin{bmatrix} w_1 \\ \vdots \\ w_{N_R} \end{bmatrix}$$

Energy per symbol

Noise PSD

Assume we know the channels at the receiver. Then we can decode the signal as

$$y = \sum_{i=1}^{N_R} h_i^* y_i = \underbrace{\sqrt{E_S} \sum_{i=1}^{N_R} | h_i |^2 s}_{\text{signal}} + \underbrace{\sqrt{N_0} \sum_{i=1}^{N_R} h_i^* w_i}_{\text{noise}}$$

and the Signal to Nose Ratio

$$SNR = \left(\sum_{i=1}^{N_R} | h_i |^2 \right) \frac{E_S}{N_0}$$

In the Wireless case the channels are random, therefore $\sum_{i=1}^{N_R} |h_i|^2$ is a random variable

Now there are two possibilities:

1. Channels strongly correlated. Assume they are all the same for simplicity

$$h_1 = h_2 = \ldots = h_{N_R} = h$$

Then

$$\sum_{i=1}^{N_R} |h_i|^2 = N_R |h|^2 = N_R \chi_2^2$$

assuming $E\left\{|h|^2\right\} = 1$

and

$$\boxed{SNR = \left(N_R |h|^2\right) \frac{E_S}{N_0} = \left(N_R \frac{1}{2} \chi_2^2\right) \frac{E_S}{N_0}}$$

From the properties of the Chi-Square distribution:

$$m_{SNR} = E\{SNR\} = N_R \frac{E_S}{N_0}$$ better on average …

$$\sigma_{SNR} = \sqrt{\mathrm{var}\{SNR\}} = \frac{N_R}{\sqrt{2}} \frac{E_S}{N_0}$$ … but with deep fades!

Define the *coefficient of variation*

$$\boxed{\mu_{\mathrm{var}} = \frac{\sigma_{SNR}}{m_{SNR}} = \frac{1}{\sqrt{2}}}$$

In this case we say that there is no diversity.

Recall the Chi-Square distribution:

1. Real Case. Let

$$y = x_1^2 + x_2^2 + \ldots + x_n^2$$

$$x_i \approx N(0,1)\ real, i.i.d.$$

Then $y = \chi_n^2$

with

$$E\{y\} = n$$

$$\mathrm{var}\{y\} = 2n$$

2. Complex Case. Let

$$y = |x_1|^2 + |x_2|^2 + \ldots + |x_n|^2$$

$$x_i = a_i + jb_i \approx CN(0,1)\ complex\ gaussian, i.i.d.$$

Then $y = \frac{1}{2}\chi_{2n}^2$

with

$$E\{y\} = n$$

$$\mathrm{var}\{y\} = \frac{1}{2}n$$

2. Channels Completely Uncorrelated.

$$SNR = \left(\sum_{i=1}^{N_R} | h_i |^2 \right) \frac{E_S}{N_0}$$

Since: $$\sum_{i=1}^{N_R} | h_i |^2 = \frac{1}{2} \chi^2_{2N_R}$$

$$\boxed{SNR = \left(\frac{1}{2} \chi^2_{2N_R} \right) \frac{E_S}{N_0}}$$

Diversity of order N_R

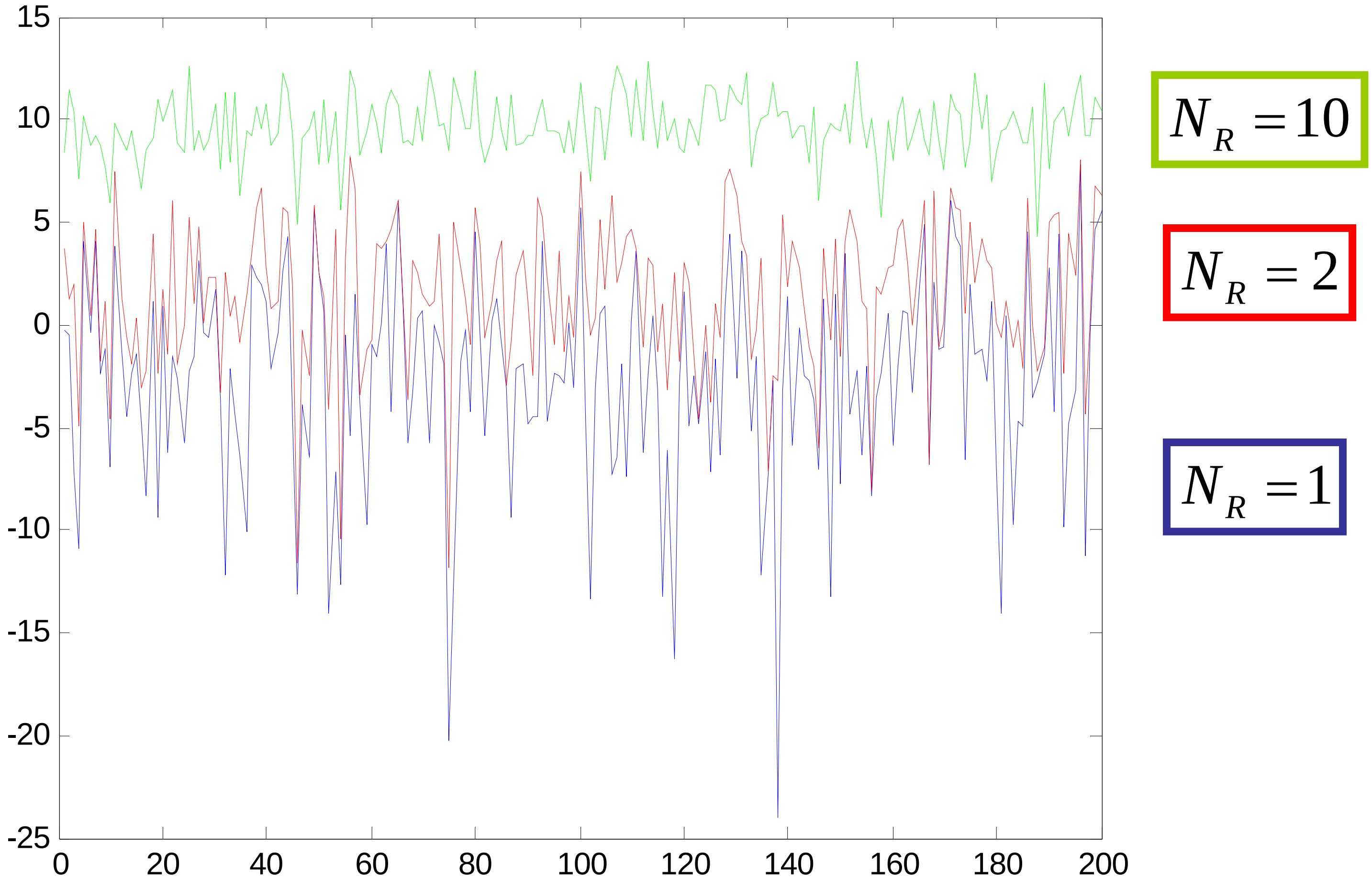
with

$$E\{SNR\} = N_R \frac{E_S}{N_0}$$

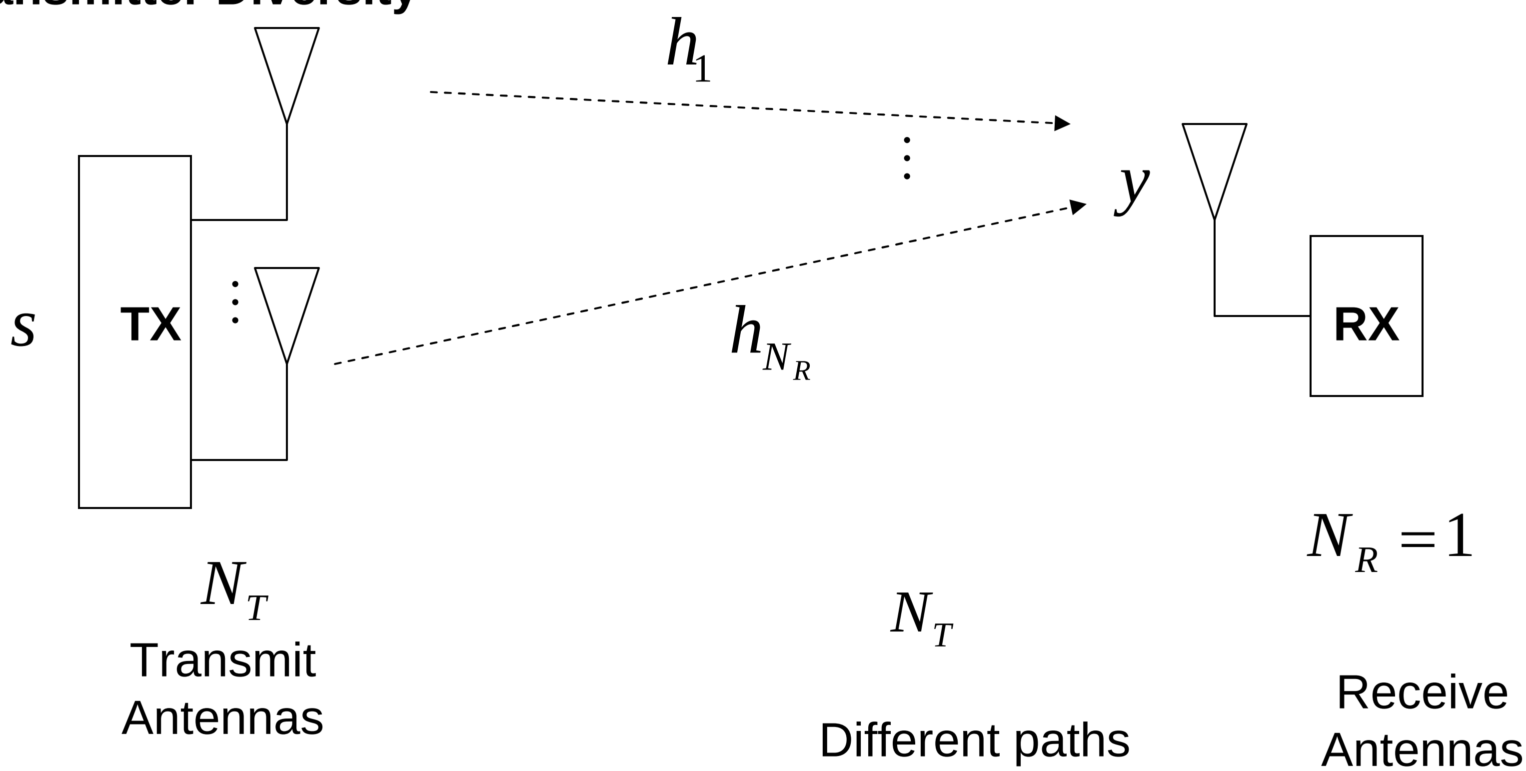
$$\sqrt{\operatorname{var}\{SNR\}} = \sqrt{\frac{N_R}{2}} \frac{E_S}{N_0}$$

$$\boxed{\mu_{\mathrm{var}} = \frac{\sigma_{SNR}}{m_{SNR}} = \frac{1}{\sqrt{2}\sqrt{N_R}}}$$

Example: overall receiver gain with receiver diversity.



Transmitter Diversity



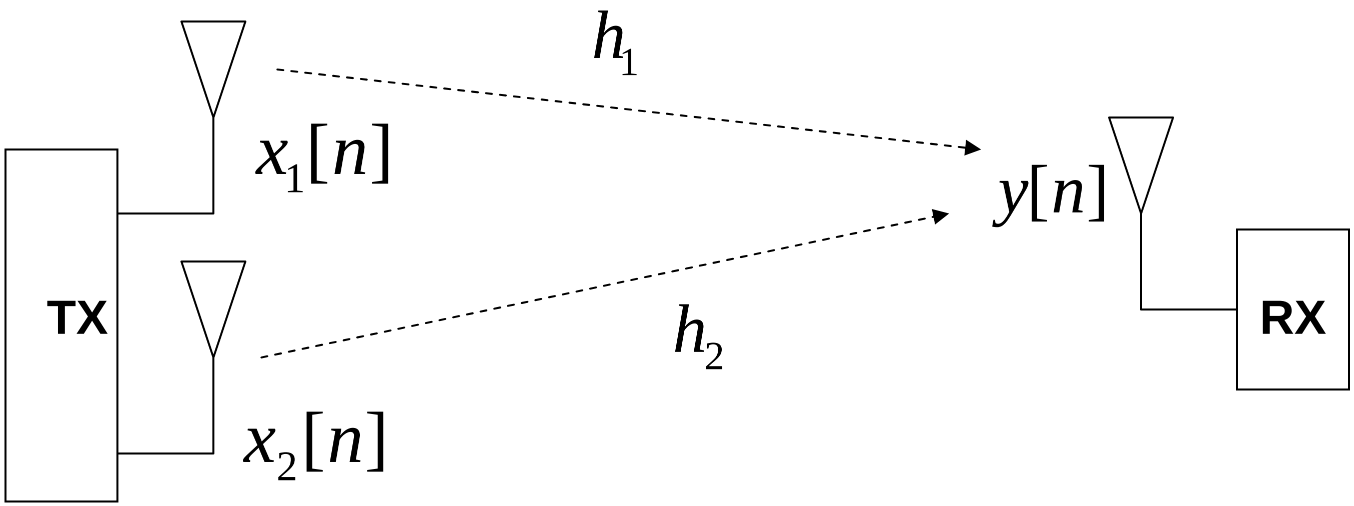
$$y = \left(\sqrt{\frac{E_S}{N_T}} \sum_{i=1}^{N_T} h_i \right) s + \sqrt{N_0}\, w$$

Equivalent to one channel,
with no benefit.

Total energy equally distributed
on transmit antennas

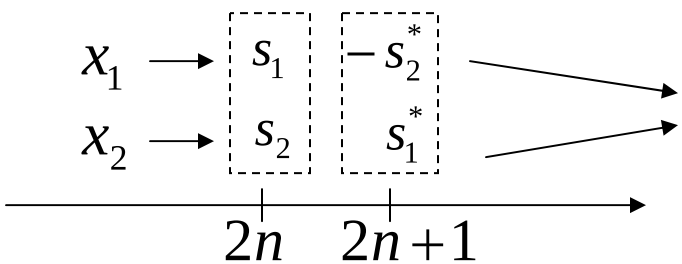
However there is a gain if we use **Space Time Coding**.

Take the case of Transmitter diversity with two antennas



Given two sequences $s_1[n], s_2[n]$

code them within the two antennas as follows



$$y[2n] = \sqrt{\frac{E_S}{2}}\left(h_1 s_1 + h_2 s_2\right) + \sqrt{N_0}\, w_1$$

$$y[2n+1] = \sqrt{\frac{E_S}{2}}\left(-h_1 s_2^* + h_2 s_1^*\right) + \sqrt{N_0}\, w_2$$

This can be written as:

$$\begin{bmatrix} y[2n] \\ y^*[2n+1] \end{bmatrix} = \sqrt{\frac{E_S}{2}} \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \sqrt{N_0} \begin{bmatrix} w_1 \\ w_2^* \end{bmatrix}$$

To decode, notice that

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} y[2n] \\ y^*[2n+1] \end{bmatrix} = \left(\sqrt{\frac{E_S}{2}} \| h \|^2 \right) \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \left(\sqrt{N_0} \| h \| \right) \begin{bmatrix} \tilde{w}_1 \\ \tilde{w}_2 \end{bmatrix}$$

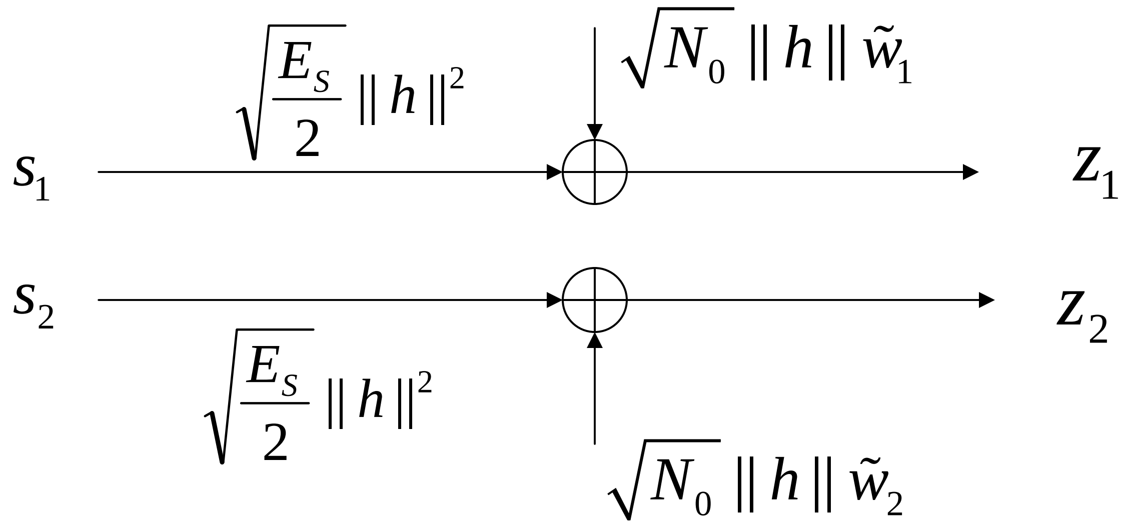
Use a Wiener Filter to estimate “s”:

$$\boxed{\begin{aligned} \hat{s}_1 &= K\left(h_1^* y[2n] + h_2 y^*[2n+1]\right) \\ \hat{s}_1 &= K\left(h_2^* y[2n] - h_1 y^*[2n+1]\right) \end{aligned}}$$

with

$$\boxed{K = \frac{\sqrt{2 / E_S}}{| h_1 |^2 + | h_2 |^2 + 2N_0 / E_S}}$$

It is like having two independent channels

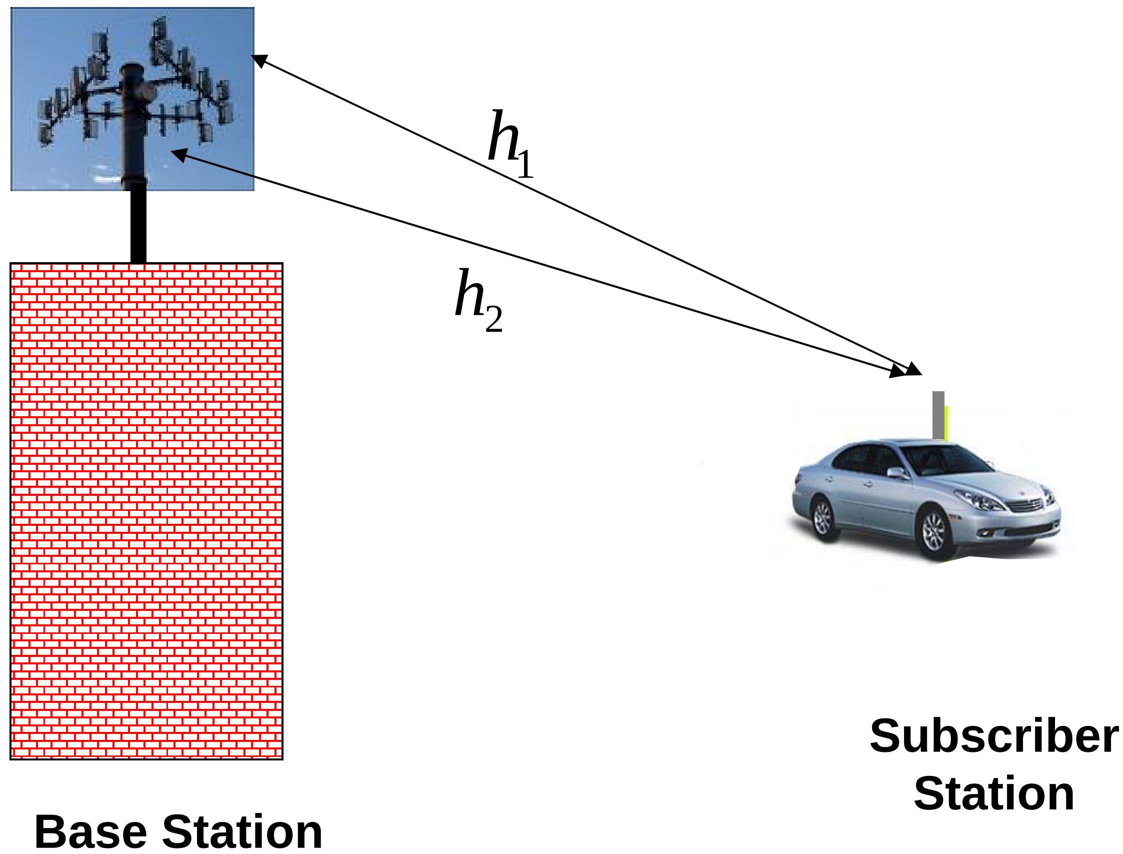


$$\boxed{SNR = \frac{\|h\|^2}{2}\frac{E_S}{N_0}}$$

$$\|h\|^2 = |h_1|^2 + |h_2|^2 = \frac{1}{2}\chi_4^2$$

Apart from the factor ½, it has the same SNR as the **receive diversity of order 2**.

WiMax Implementation



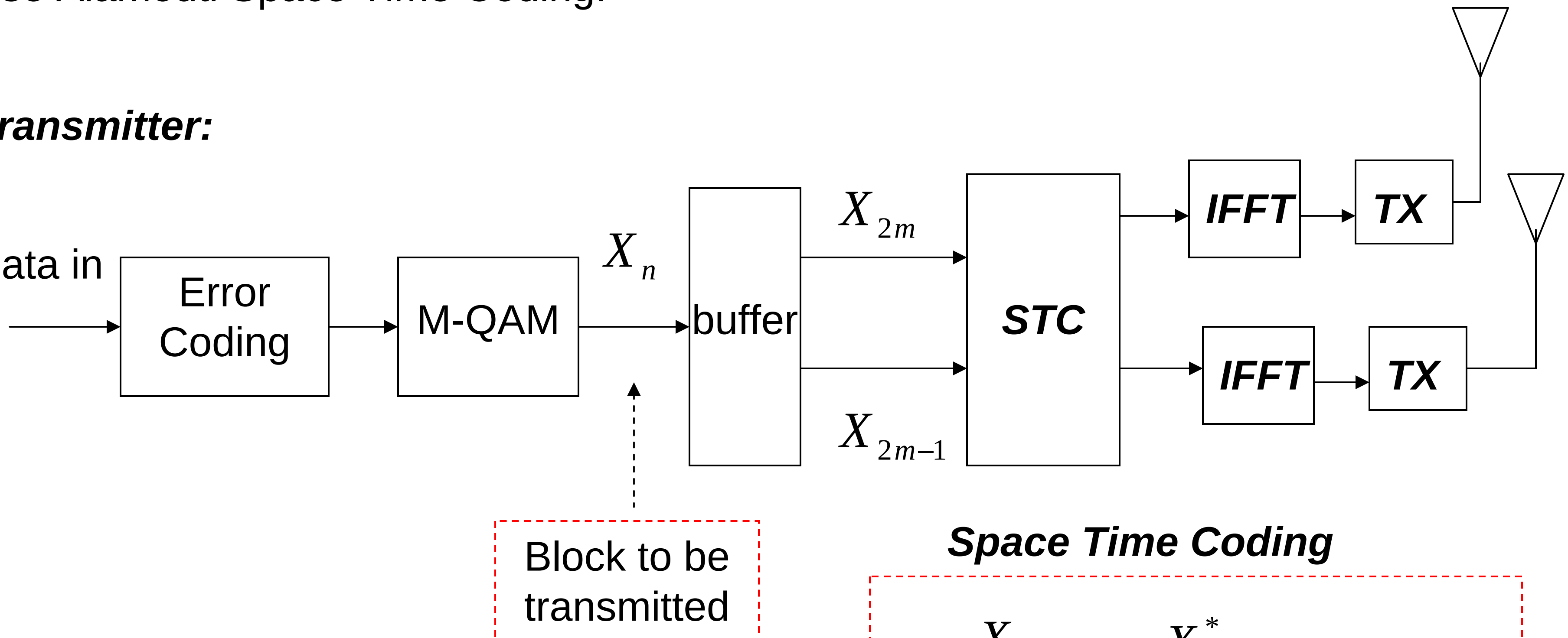
Down Link (DL): BS -> SS Transmit Diversity

Uplink (UL): SS->BS Receive Diversity

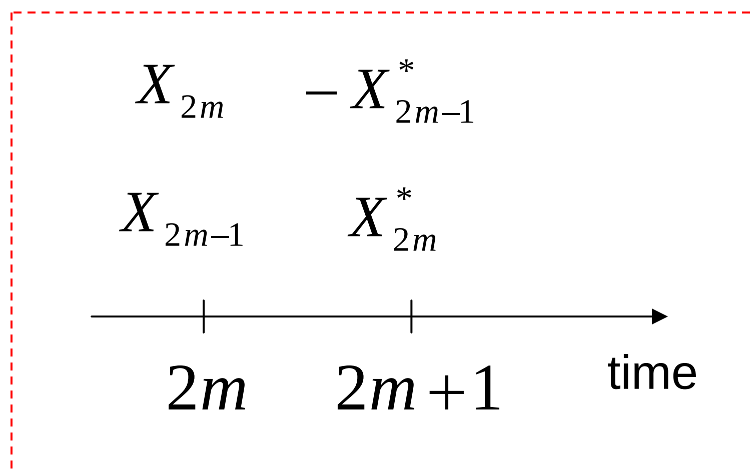
Down Link: Transmit Diversity

Use Alamouti Space Time Coding:

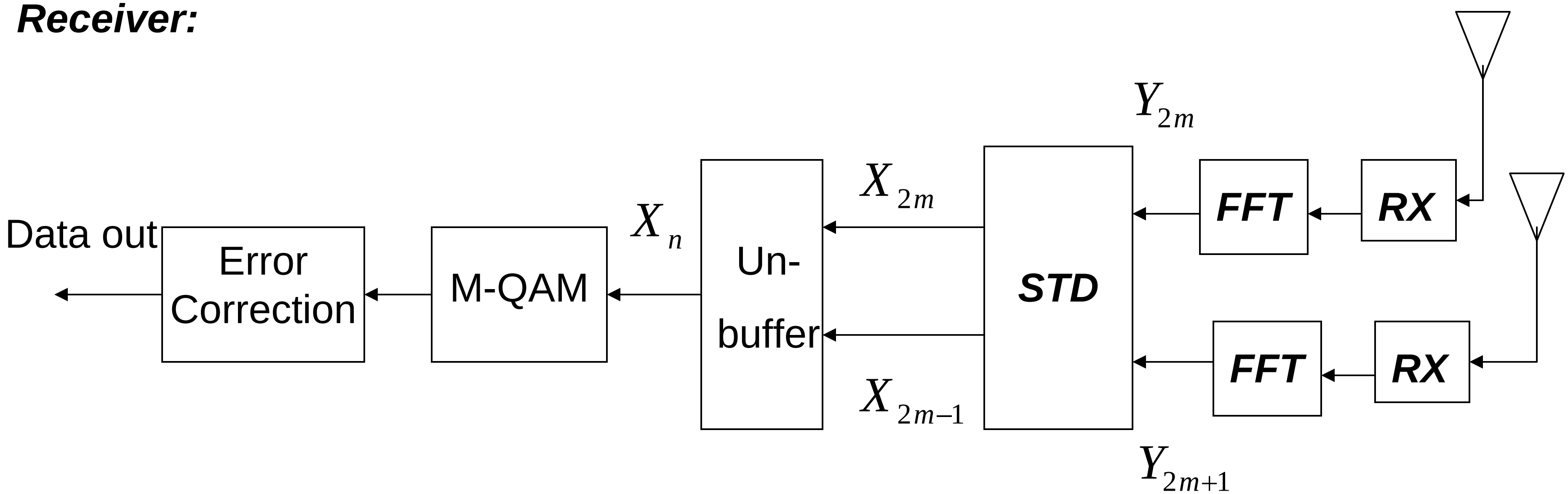
Transmitter:



Space Time Coding



Receiver:



Space Time Decoding:

For each subcarrier *k* compute:

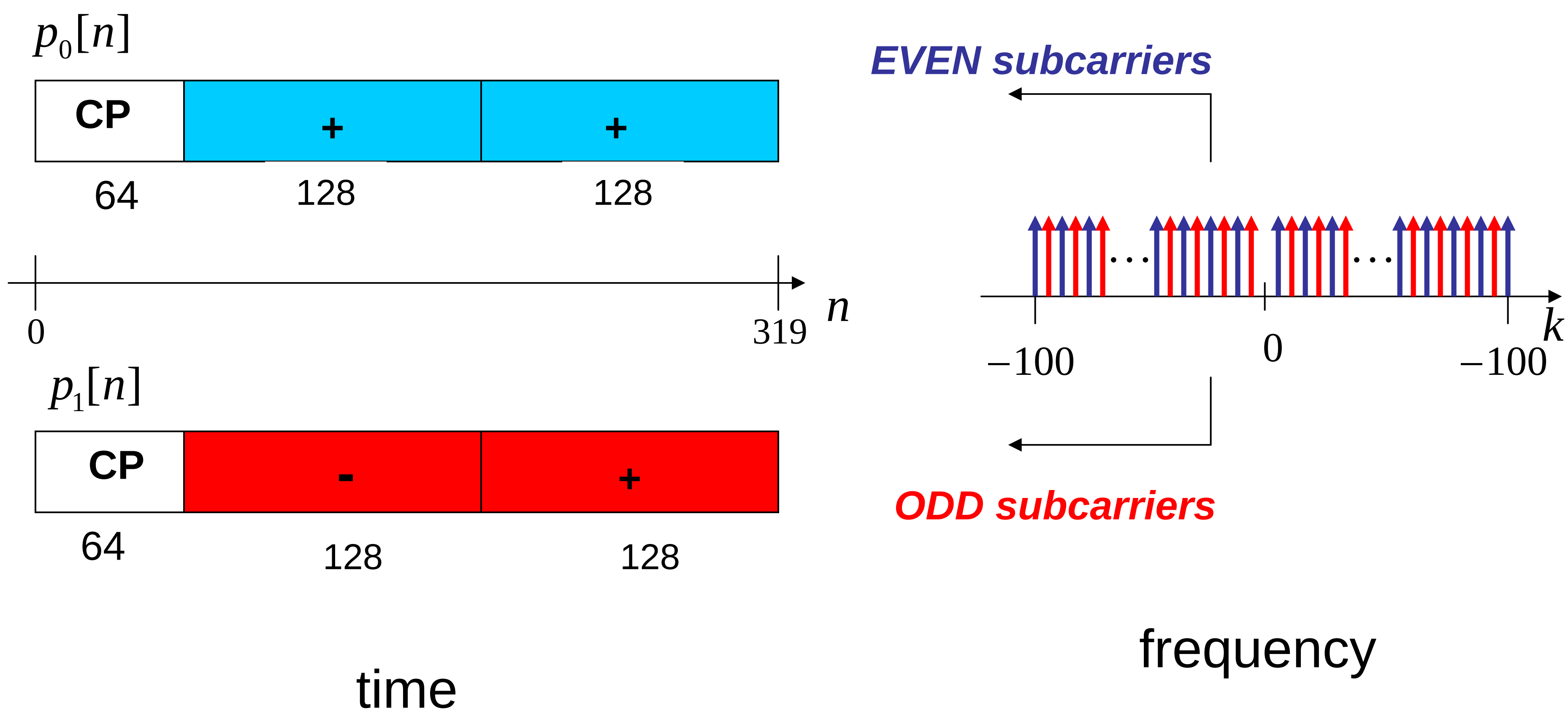
$$\boxed{\begin{aligned}\hat{X}_{2m}[k] &= K\left(H_1^*[k]Y_{2m}[k] + H_2[k]Y_{2m+1}^*[k]\right)\\ \hat{X}_{2m-1}[k] &= K\left(H_2^*[k]Y_{2m}[k] - H_1[k]Y_{2m+1}^*[k]\right)\end{aligned}}$$

with

$$\boxed{K = \frac{\sqrt{2/E_S}}{|H_1[k]|^2 + |H_2[k]|^2 + 2N_0/E_S}}$$

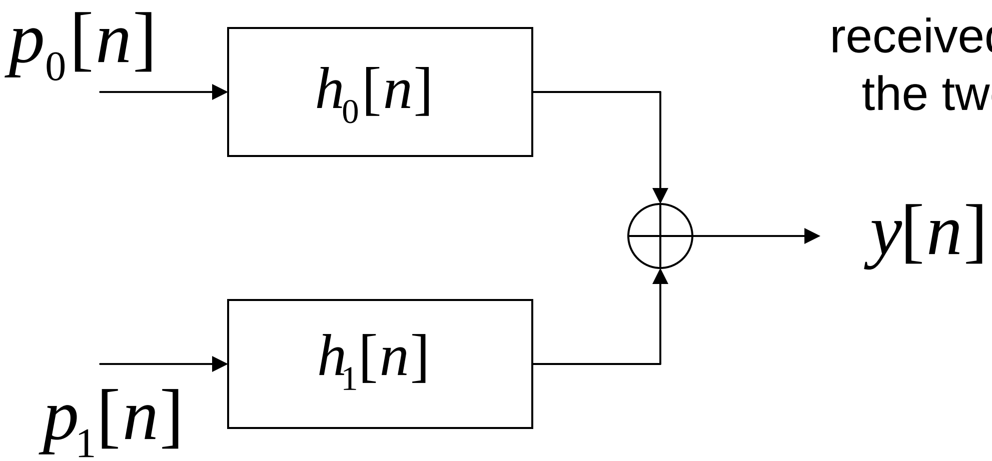
Preamble, Synchronization and Channel Estimation with Transmit Diversity (DL)

The two antennas transmit two preambles at the same time, using different sets of subcarriers



Both preambles have a symmetry:

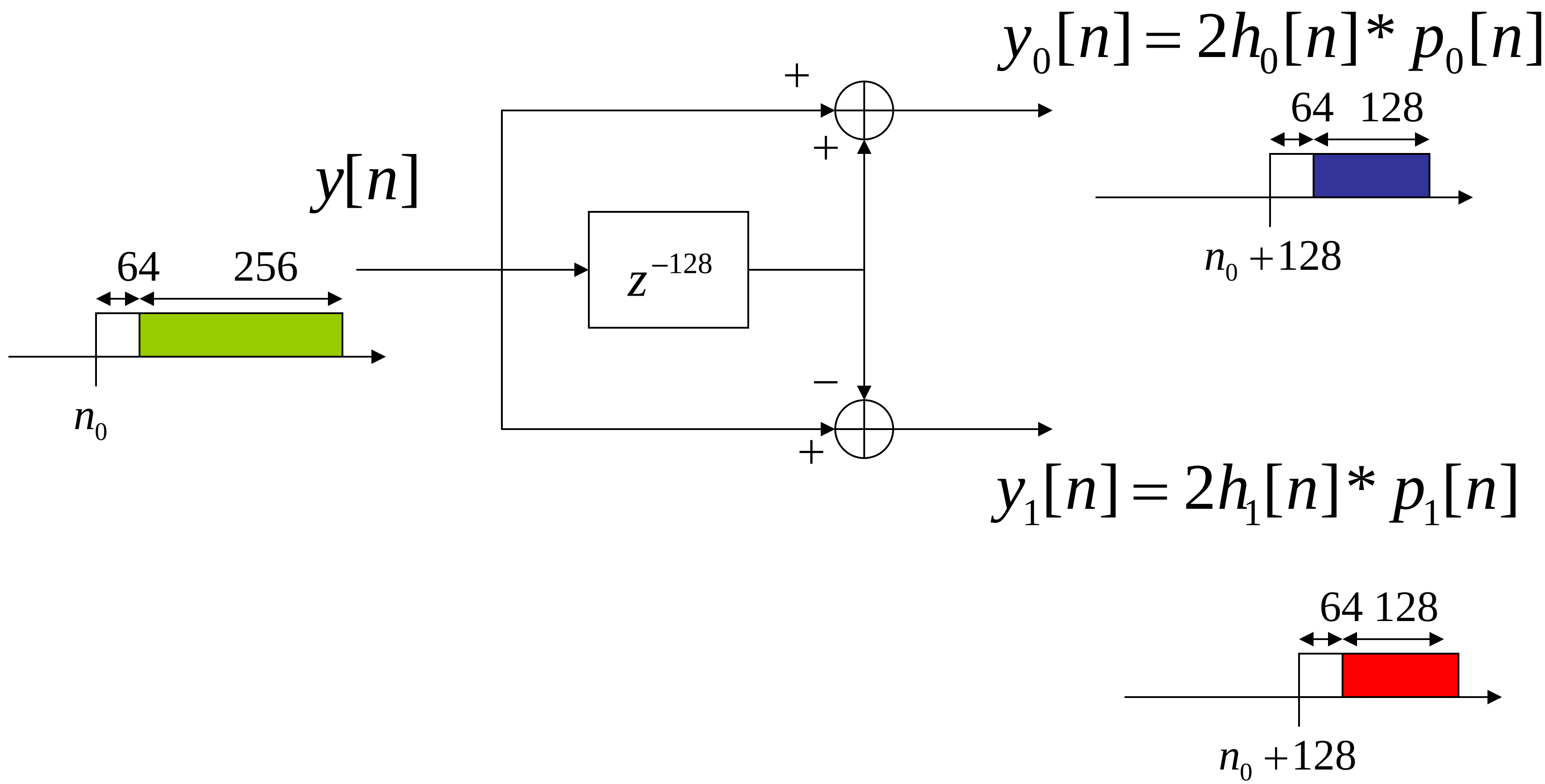
$$p_0[n] = p_0[n-128]$$
$$p_1[n] = -p_1[n-128] \qquad n = 128,\ldots,319$$



Problems:

- time synchronization
- estimation of both channels

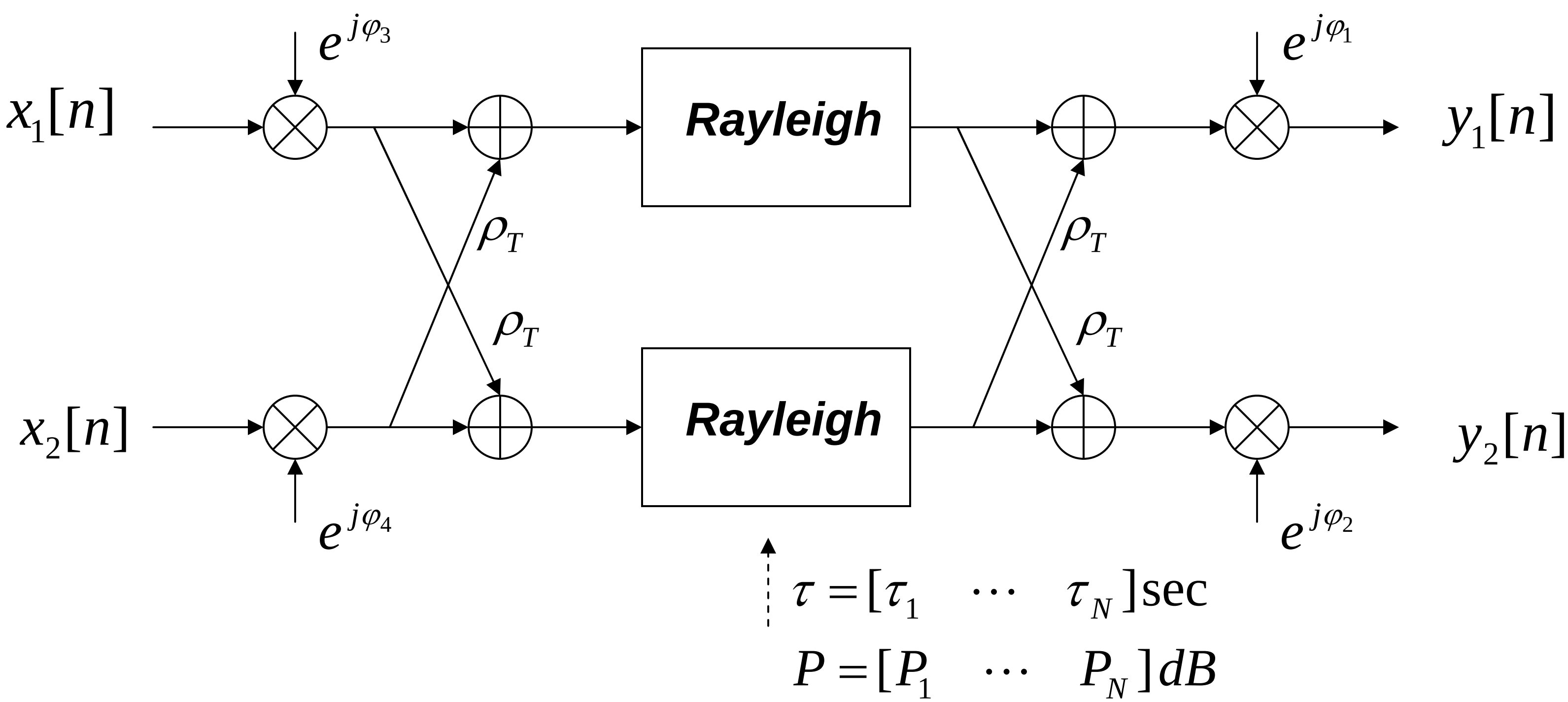
One possibility: use symmetry of the preambles



The two preambles can be easily separated

MIMO Channel Simulation

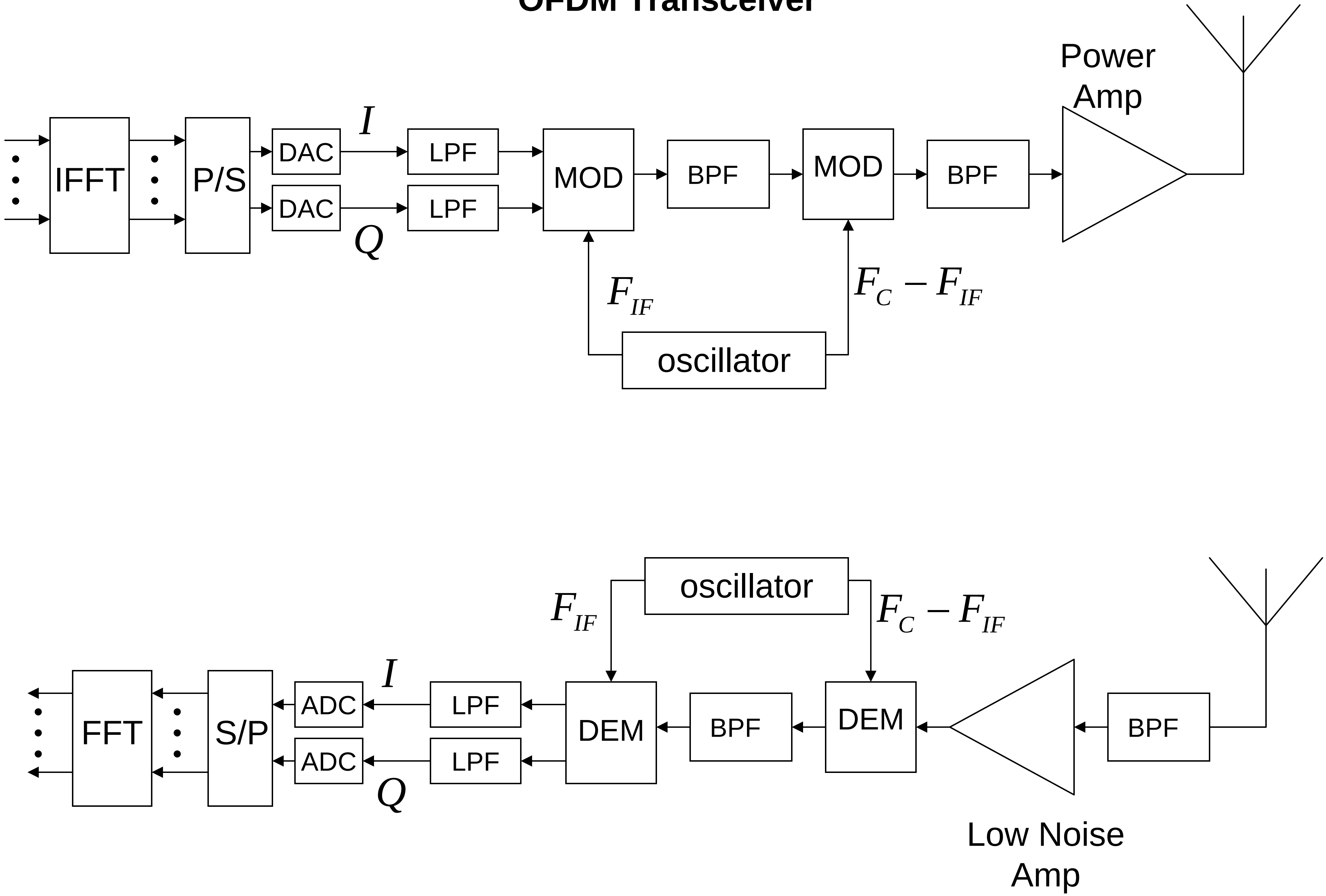
Take the general 2x2 channel



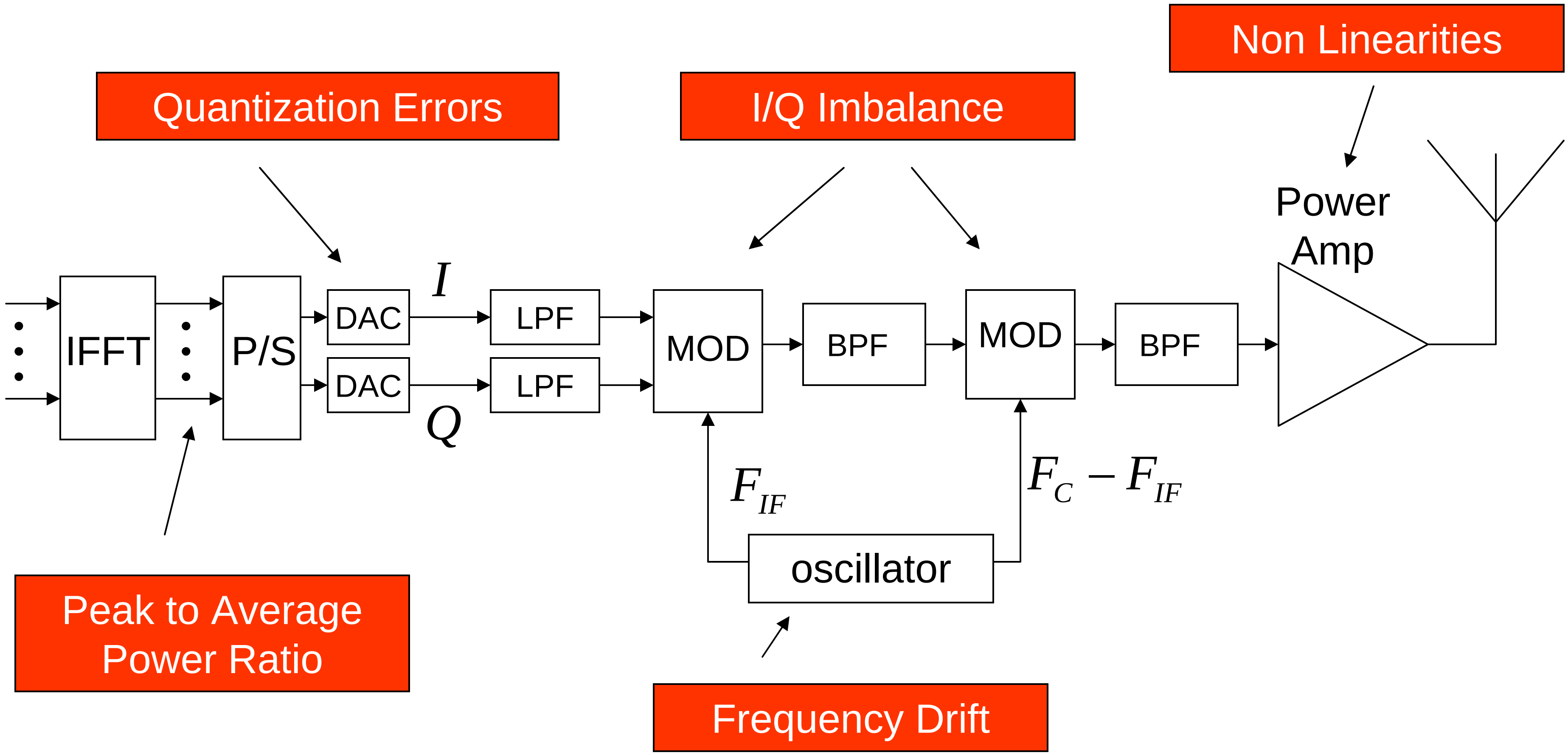
$0 \le \rho_T \le 1$ Correlation at the transmitter

$0 \le \rho_R \le 1$ Correlation at the receiver

OFDM Transceiver



Problems!!!



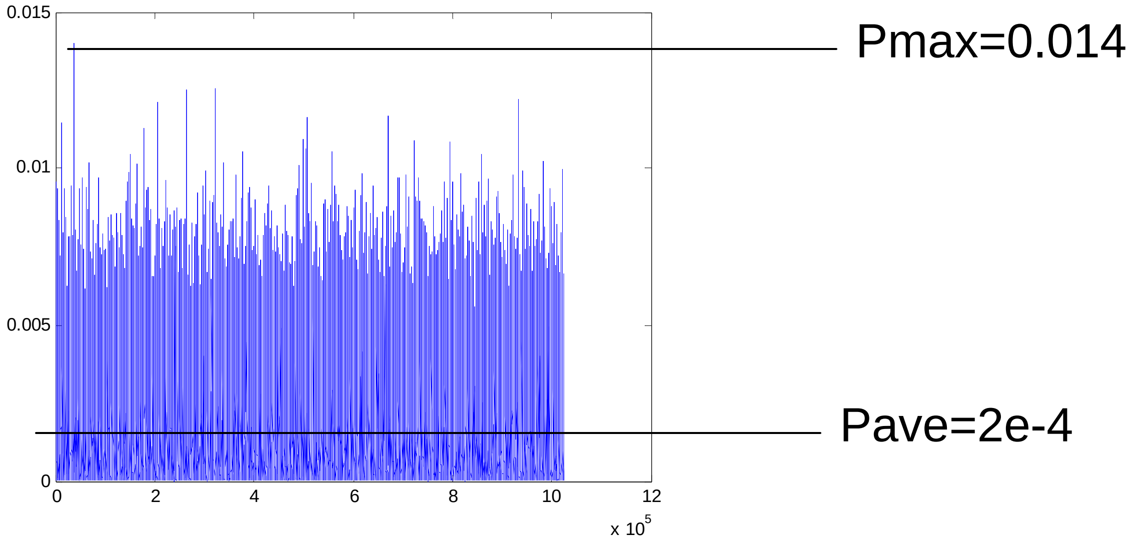
Non Linearities
Quantization Errors
I/Q Imbalance
Power Amp
IFFT
P/S
DAC
DAC
I
Q
LPF
LPF
MOD
BPF
MOD
BPF
F_{IF}
$F_C - F_{IF}$
oscillator
Peak to Average Power Ratio
Frequency Drift

• **Peak to Average Power Ratio (PAR)**

A Single Carrier QPSK signal has a constant amplitude, since only the phase is modulated.

A Multi Carrier OFDM signal has a random amplitude.

<u>Example</u>: see an OFDM signal with 1024 subcarriers. Plot 1,000 OFDM symbols



Pmax/Pave=66.88=18dB

Effects:

- On the ADC: larger number of bits to accommodate large dynamics
- On the Power Amplifier: it has to be linear within a wider range

Remedies:

- the signal needs to be clipped to avoid saturating the amplifier
- the output power of the signal has to be reduced

Cause of large peaks:

OFDM signal is a sum of sinusoids. When all in phase, the sum can be large (peak value) with respect to the average.

$$y[n]=\frac{1}{N}\sum_{k=0}^{N-1}Y[k]e^{jk\frac{2\pi}{N}n}$$

Assume all subcarriers Y[k] are i.i.d.:

Average Power: $$E\left\{\| y[n]\|^2\right\}=\left|\frac{1}{N^2}\sum_{k=0}^{N-1}E\left\{\| Y[k]\|^2\right\}\right|=\frac{1}{N}E\left\{\| Y[k]\|^2\right\}$$

Peak Power when all subcarriers are aligned: $$E\left\{\| y[n]\|^2 \;\middle|\; Y[k]=|Y|\right\}\leq |Y|^2$$

$$\frac{P_{Peak}}{P_{Ave}}\propto N$$ It increases with the number of subcarriers

How bad is PAR?

Probability Distribution of PAR

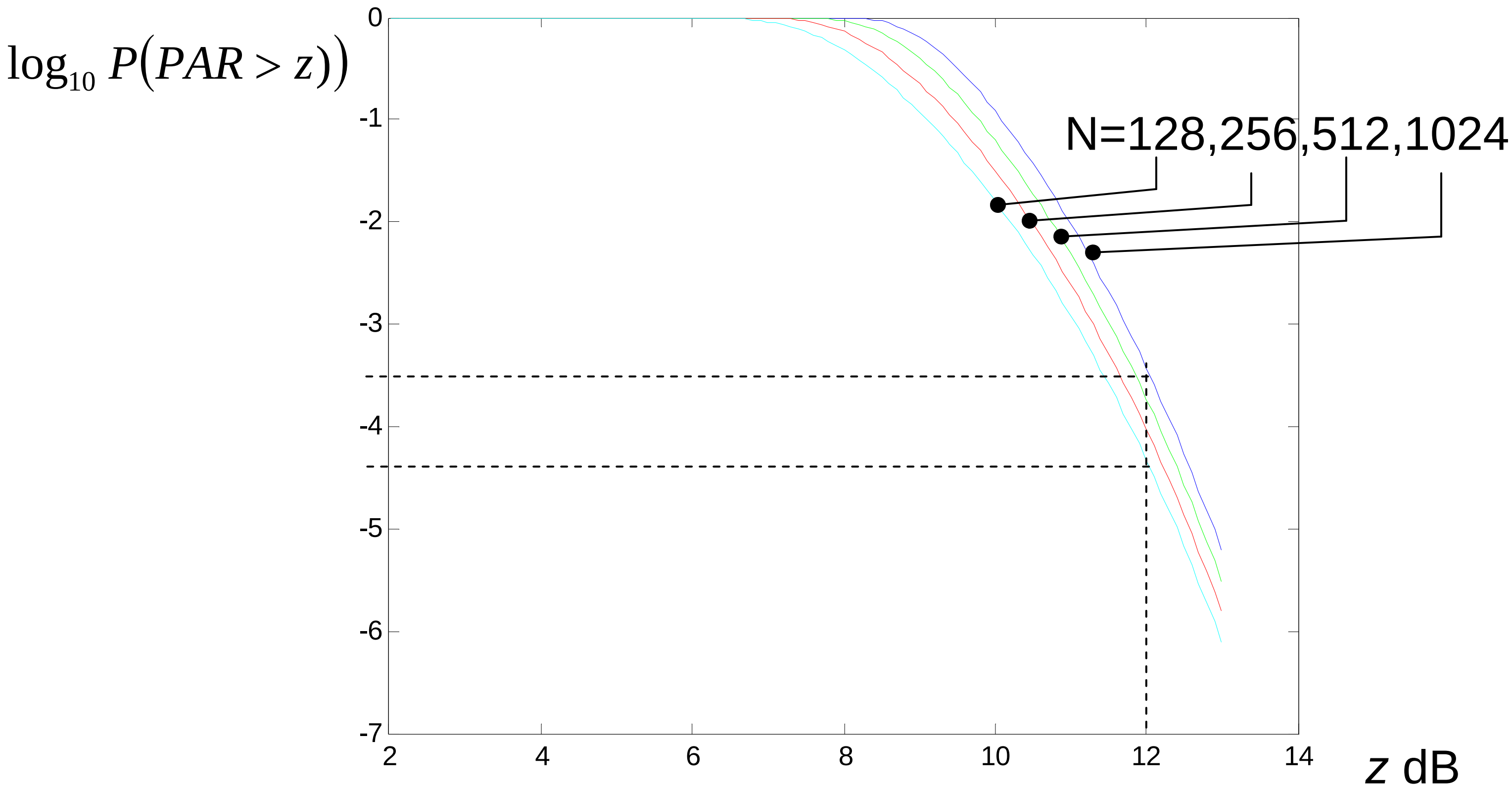
By the Central Limit Theorem the received signal $y[n]$ is gaussian

The amplitude is Rayleigh distributed and the power is Chi-square.

Therefore we can compute the Cumulative Density Function analytically:

$$\Pr\left\{PAR > z\right\} = \left(1 - \left(1 - e^{-z}\right)^{2.8N}\right)$$

fairly accurate for number of carriers $N \geq 64$



Notice:

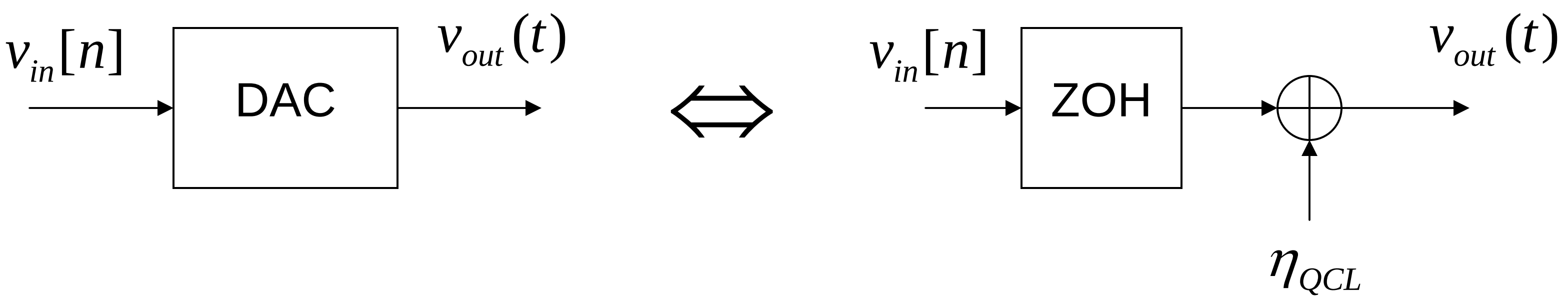
1. The probability that PAR=N, the number of subcarriers, is almost zero;
2. The probability that PAR>4=12dB is of the order of $10^{-4.5} - 10^{-3.5}$

Remedies:

1. <u>Data randomizing</u>: if there is an error due to PAR, resend the data and most likely it will be fine
2. <u>Clipping</u>: easy thing to do, at the expenses of introducing errors
3. <u>Coding</u>: potentially the best, but still not reliable solutions (not in the standard)
4. A number of <u>iterative</u> techniques: too complex for real time implementation

Effects of PAR on Digital to Analog Converters (DAC)

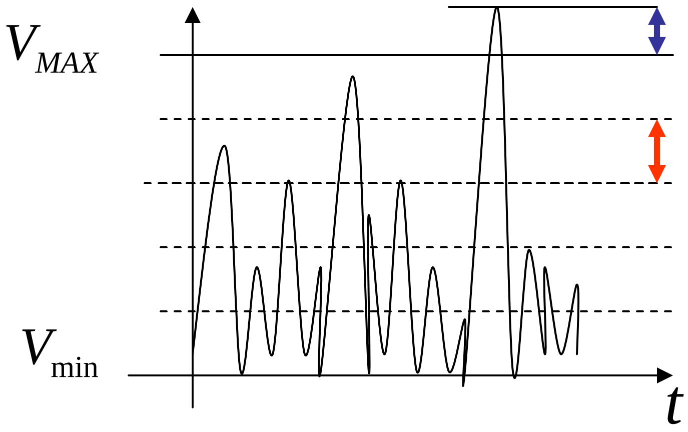
For a fixed number of bits, we need to compromise between two sources of errors:



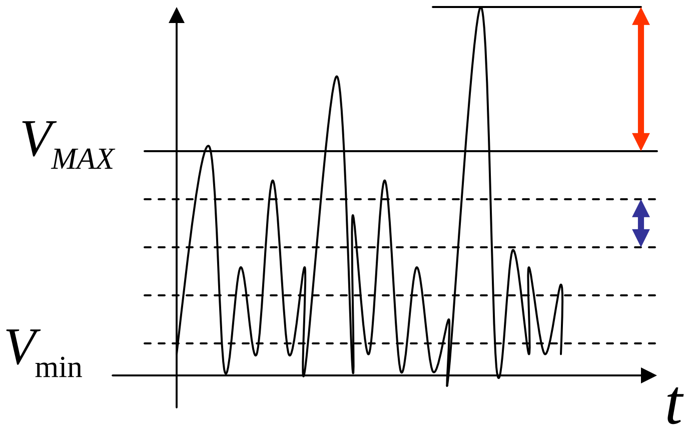
In the DAC there are two sources of errors:

- Quantization Error;
- Saturation error due to clipping

Compromise between Number of Bits in DAC and Clipping Level

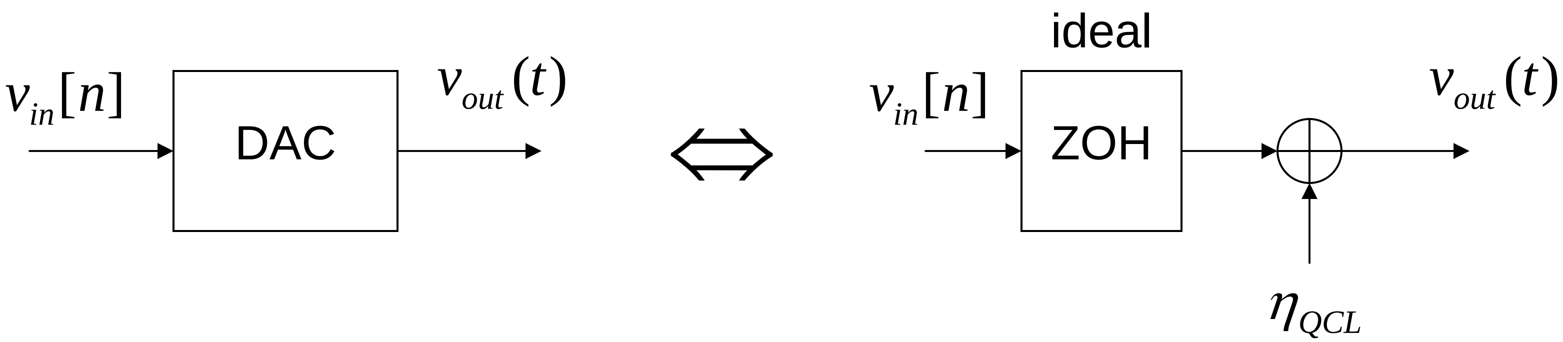


Small clipping error, Large Quantization Noise



Large clipping error, Small Quantization Noise

We choose Clipping Level for best compromise between clipping error and quantization noise.



Total SNR due to Q and CL: $SNR_{QCL} = \left(SNR_Q^{-1} + SNR_{CL}^{-1}\right)^{-1}$

SNR due to Q : $SNR_Q = \dfrac{12 \times 2^{2b}}{(2\mu)^2}$

SNR due to CL : $SNR_{CL} = \sqrt{\frac{\pi}{8}}\mu^3 e^{\mu^2/2}$

where

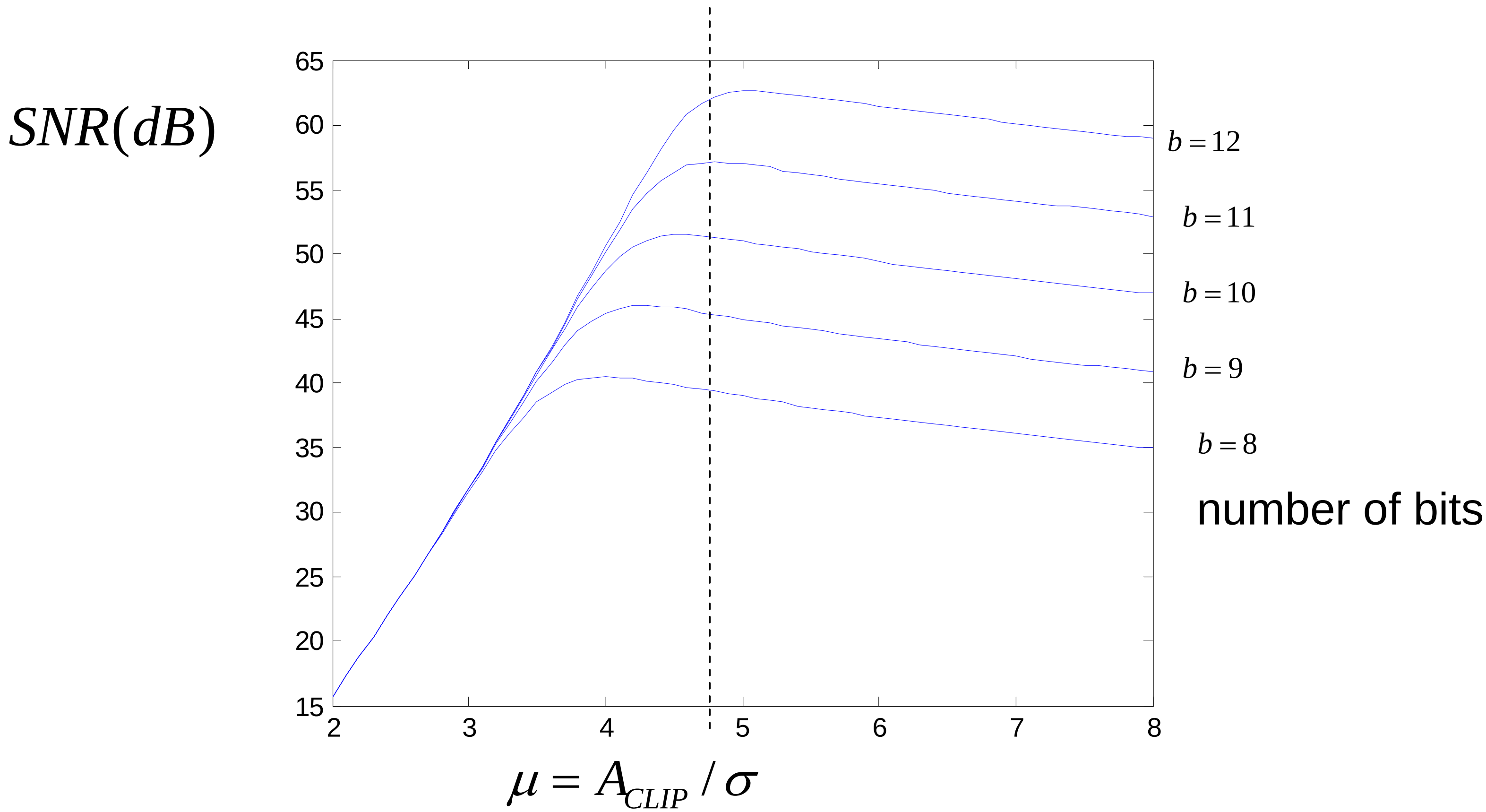
$$\mu = A_{CLIP} / \sigma$$

$$\sigma = \sqrt{E\left\{|x[n]|^2\right\}}$$

b = number of bits per sample

See both combined:

MAX around $\mu = A_{CLIP}/\sigma = 4-5$



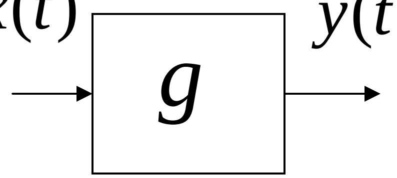
Choose max clipping value: $V_{MAX} \approx 4\sqrt{E\left\{\left|v_{in}[n]\right|^2\right\}}$

We have seen that the probability that PAR>4=12dB is very small.

Effects of PAR on Power Amplifiers

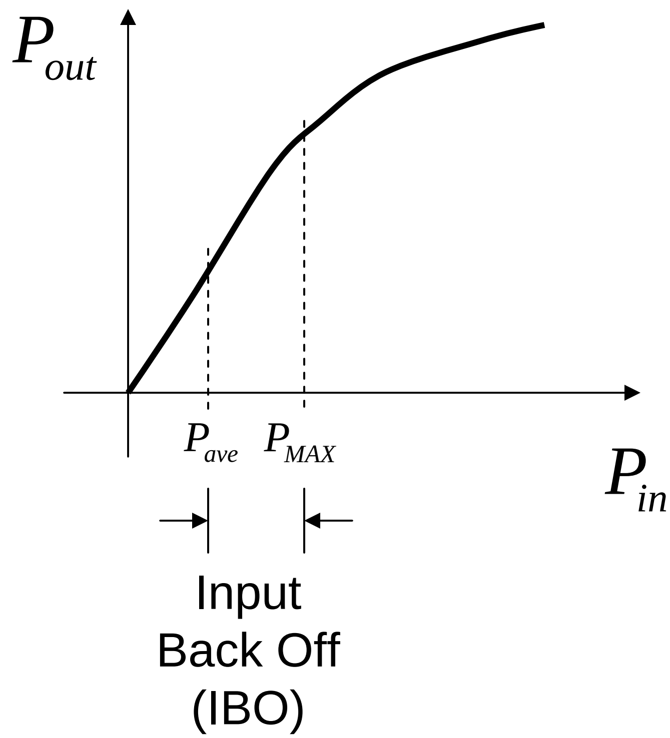
Rapp's model of a Power Amplifier:

$$x(t) \qquad y(t) = g\big(x(t)\big)$$



$$g(x) = \frac{1}{\left(1 + x^{2p}\right)^{\frac{1}{2p}}}$$

p of the order of 3



Ideally you want the Input Back Off (IBO) to be larger than the PAR.

$$IBO = 10\log_{10}\frac{P_{MAX}}{P_{ave}} > PAR$$

$$PAR \leq 10\log_{10} N$$

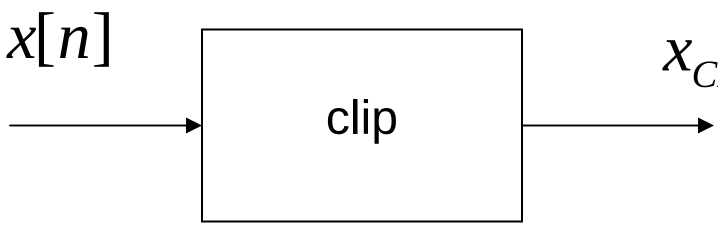
This can be excessive and very inefficient.

<u>Example</u>: with *N=256* carriers you need a IBO of

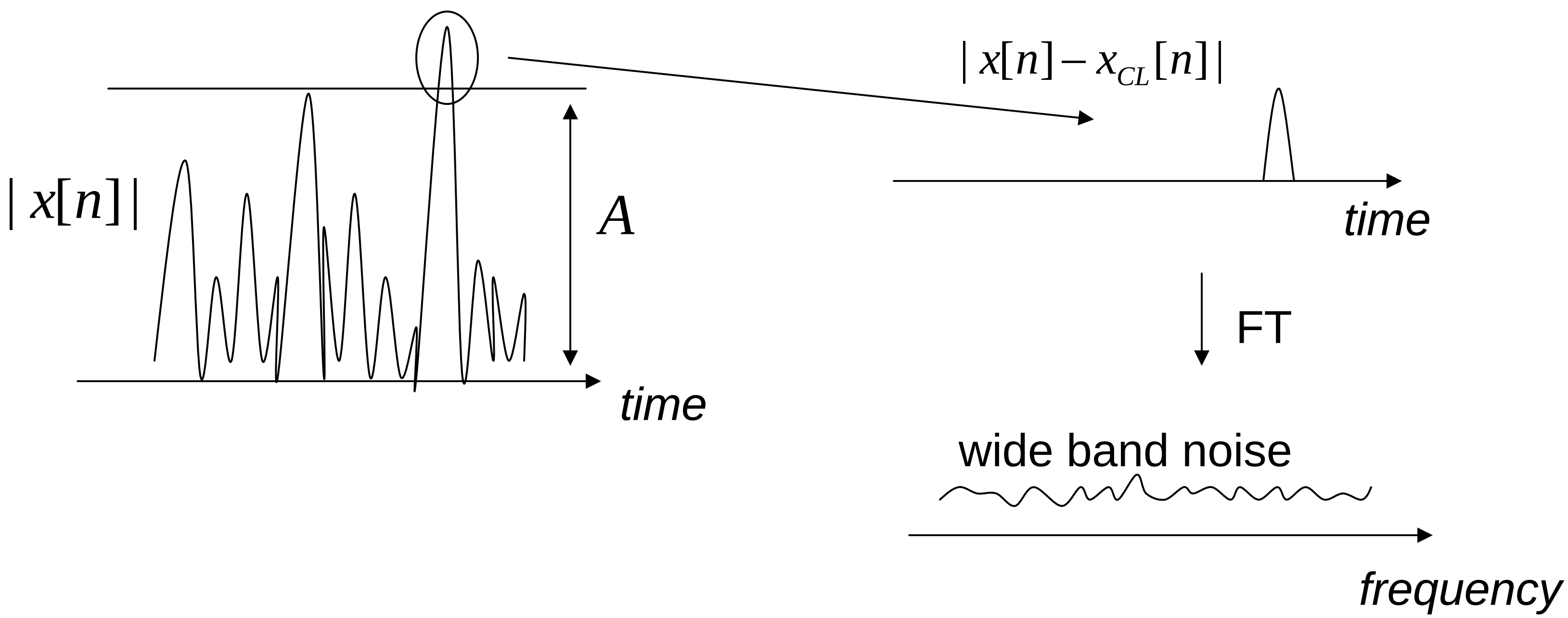
$$IBO = PAR = 256 = 24dB$$

to guarantee that the amplifier always operates in the linear region.

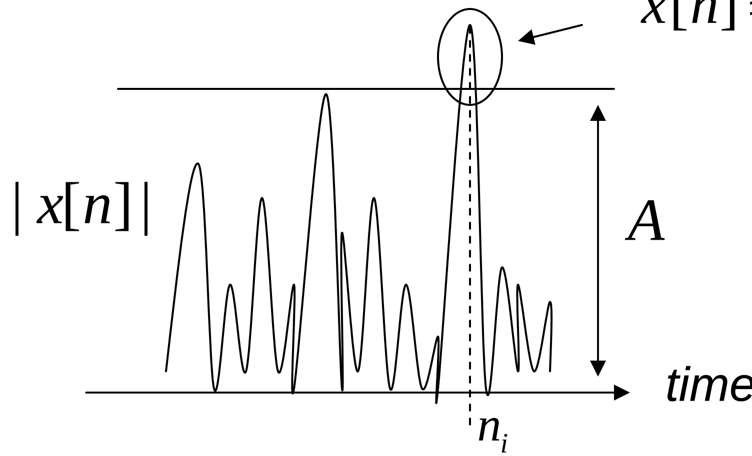
Remedies: Clipping



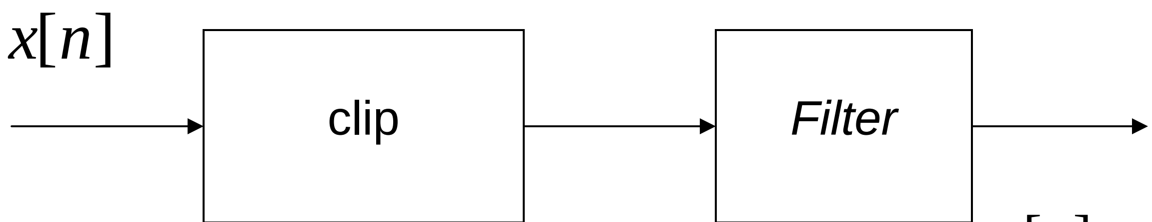
$$x_{CL}[n] = \begin{cases} Ae^{j\angle x[n]} & \text{if } |x[n]| > A \\ x[n] & \text{if } |x[n]| \le A \end{cases}$$



$$\tilde{x}[n] = \sum_i a_i e^{j\phi_i}\delta[n-n_i]$$

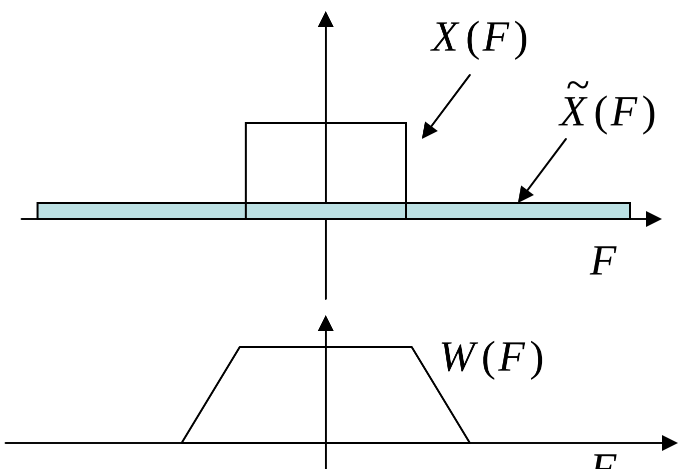


$$x_{CL}[n] = x[n] - \tilde{x}[n]$$



$$x_W[n] = x[n] * w[n] - \sum_i a_i e^{j\phi_i} w[n-n_i]$$

$$\cong x[n] - \sum_i a_i e^{j\phi_i} w[n-n_i]$$



It has two effects:

1. Additive Noise in the demodulated subcarriers
2. It generates out of band noise, so we need to add a Low Pass Filter

The latter is not allowed, since it interferes with neighboring channels.

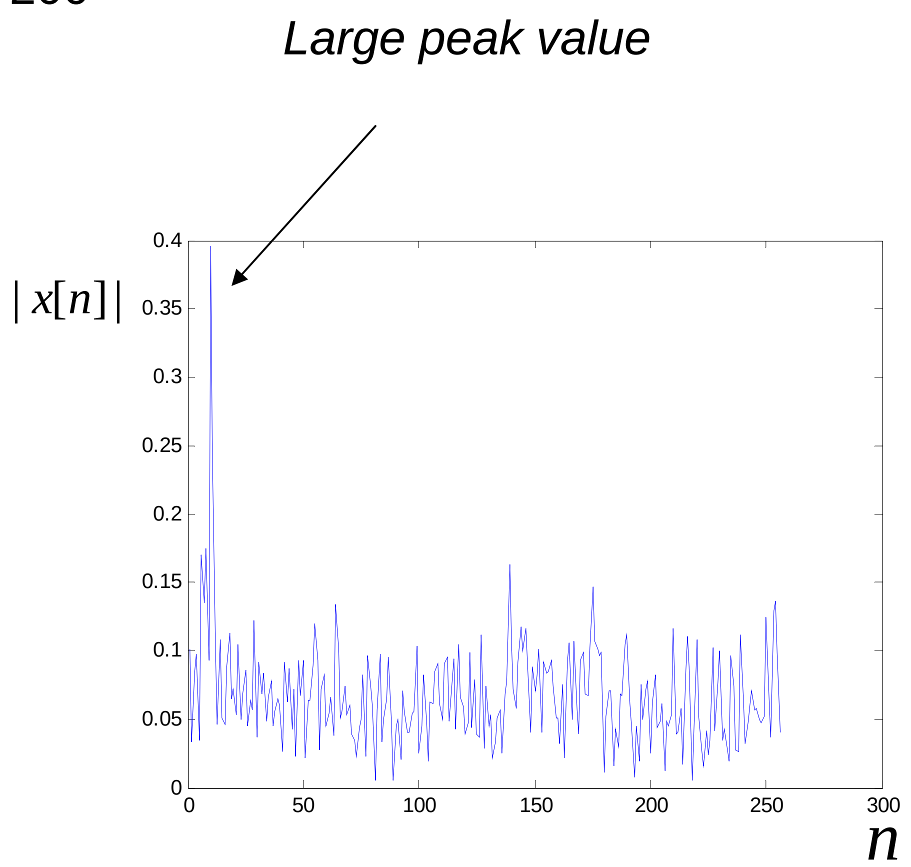
Example:

See one of OFDM symbol with the following parameters (this is not random but chosen with high PAR):

FFT length N=256

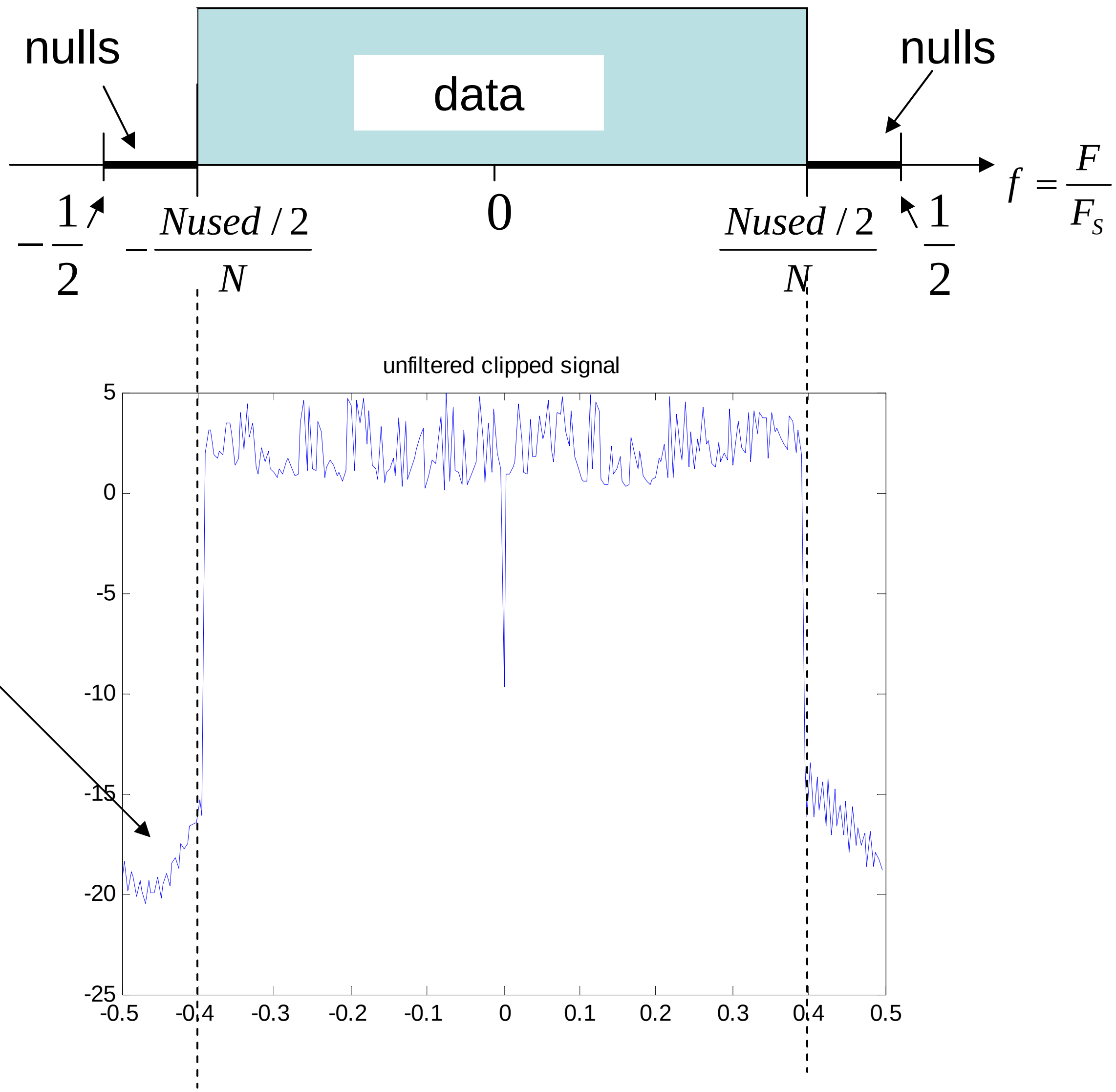
Data Carriers: Nused=200

Modulation: 4QAM



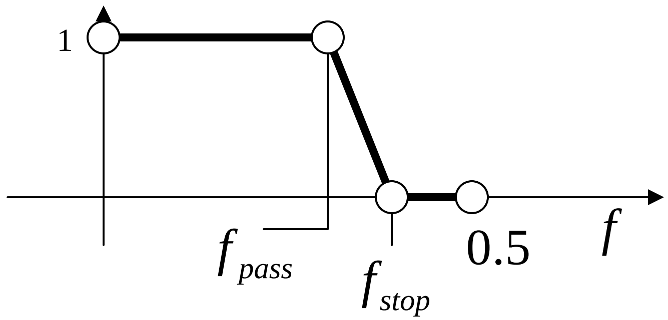
Clip it to 5dB (no filtering):

See the Frequency Spectrum:



Large values in the freq. guards

Choose a filter with this characteristics:



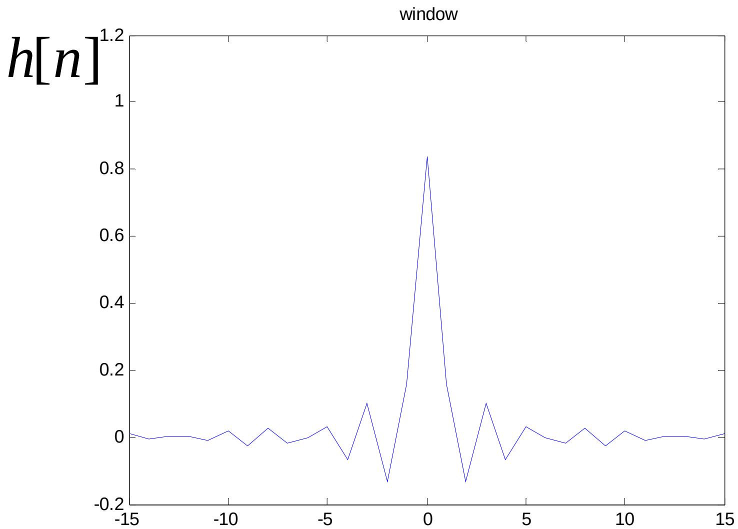
$$f_{pass} = Nused/(2N)$$

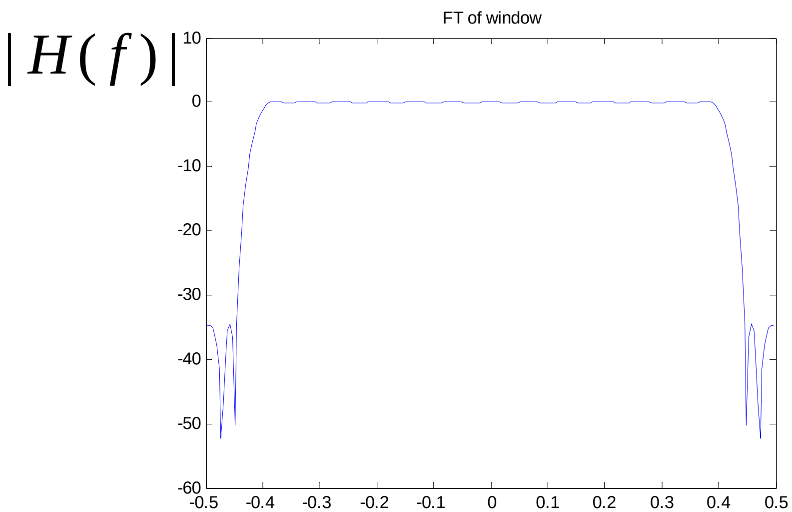
$$f_{pass} < f_{stop} < 1/2$$

In matlab: use *firpm*

```
h=firpm(M, [0,fpass,fstop,0.5]*2, [1,1,0,0]);
```

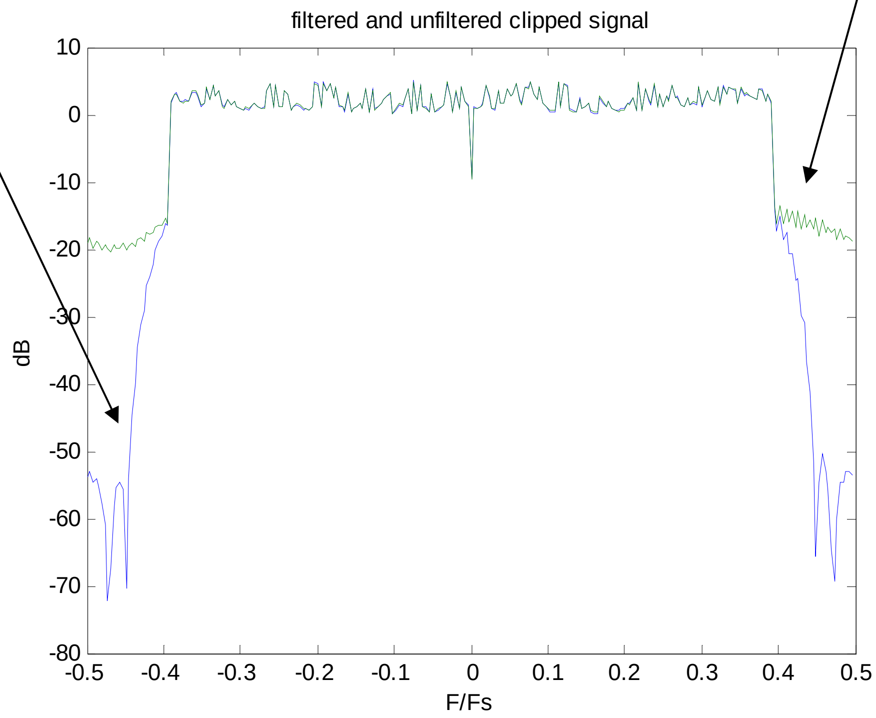
filter order, say M=40



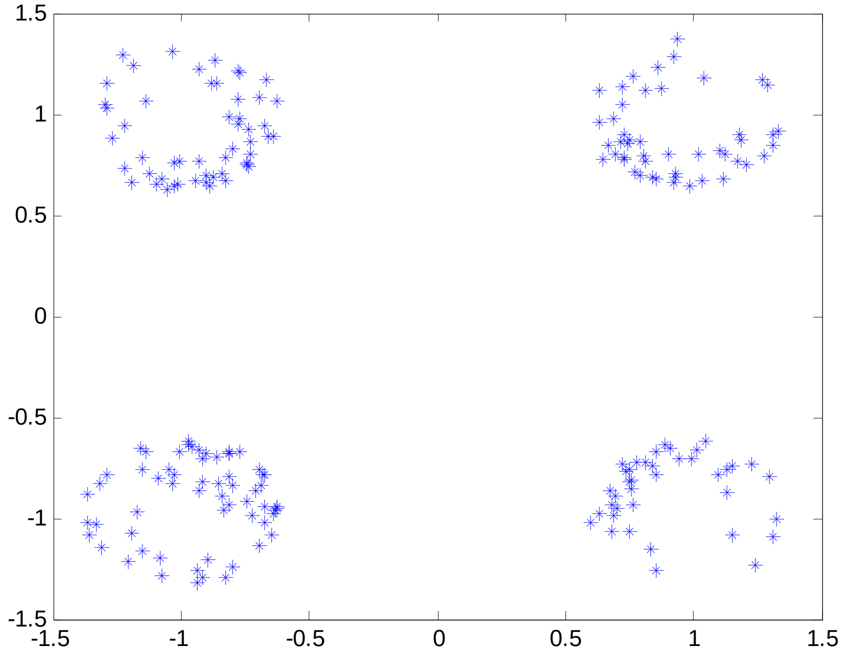


Clipped and Filtered:

Clipped only (no filter)

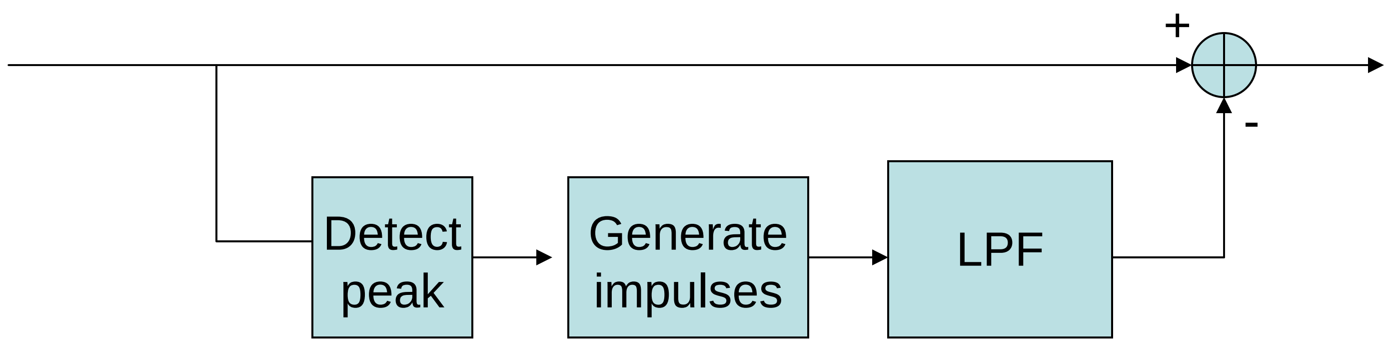


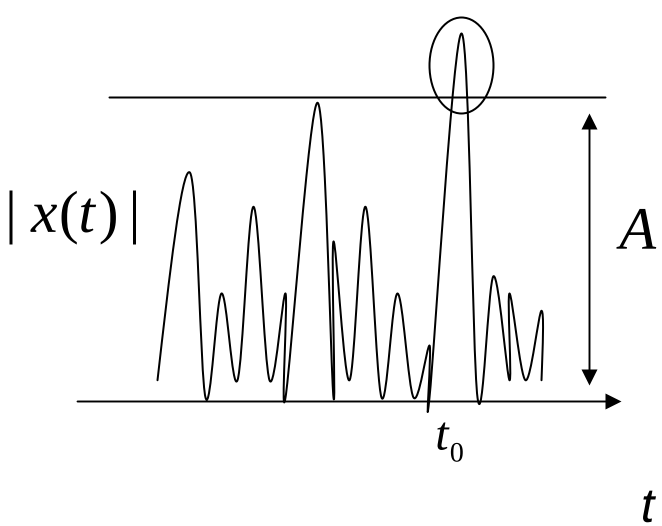
Negative effect on data!!!

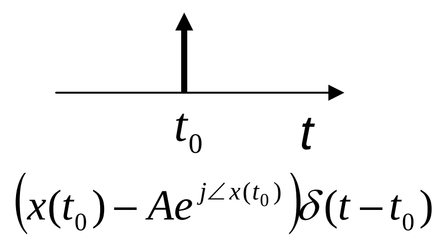


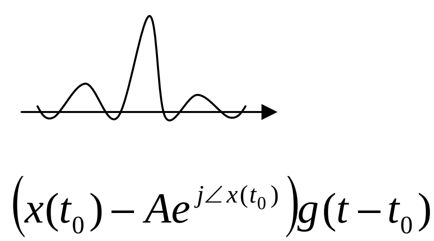
Peak Cancellation

Detect and Cancel the peak to reduce out of band noise.









It spreads the error to neighboring values.

Larger in band noise

Effects:

- The LPF completely controls the out of band noise. It can be completely eliminated;
- It reduced PAR and therefore the Input Back Off.
- It causes an increase of in band noise and therefore a higher BER.

Effect of I/Q Imbalance

Ideally the received signal is of the form

$$y[n] = \sum_{k=0}^{N-1} Y[k] e^{jk\frac{2\pi}{N}n} = y_I[n] + j y_Q[n]$$

In practice the I and Q components have different gains, as

$$\begin{aligned}\bar{y}[n] &= A_I e^{j\varphi_I} y_I[n] + j A_Q e^{j\varphi_Q} y_Q[n] \\ &= \bar{A} e^{j\bar{\varphi}} \left((1+\delta) e^{j\varepsilon} y_I[n] + j(1-\delta) e^{-j\varepsilon} y_Q[n] \right)\end{aligned}$$

with $\bar{A} = \dfrac{A_I + A_Q}{2}$ $\qquad \delta = \dfrac{A_I - A_Q}{A_I + A_Q}$

$\bar{\varphi} = \dfrac{\varphi_I + \varphi_Q}{2}$ $\qquad \varepsilon = \dfrac{\varphi_I - \varphi_Q}{\varphi_I + \varphi_Q}$

The two parameters δ, ε express the magnitude and phase of the I/Q imbalance.

Assume them small:

$$(1+\delta)e^{j\varepsilon} \cong (1+\delta)(1+j\varepsilon) \cong 1+\delta+j\varepsilon$$
$$(1-\delta)e^{-j\varepsilon} \cong (1-\delta)(1-j\varepsilon) \cong 1-\delta-j\varepsilon$$

After a bit of algebra:

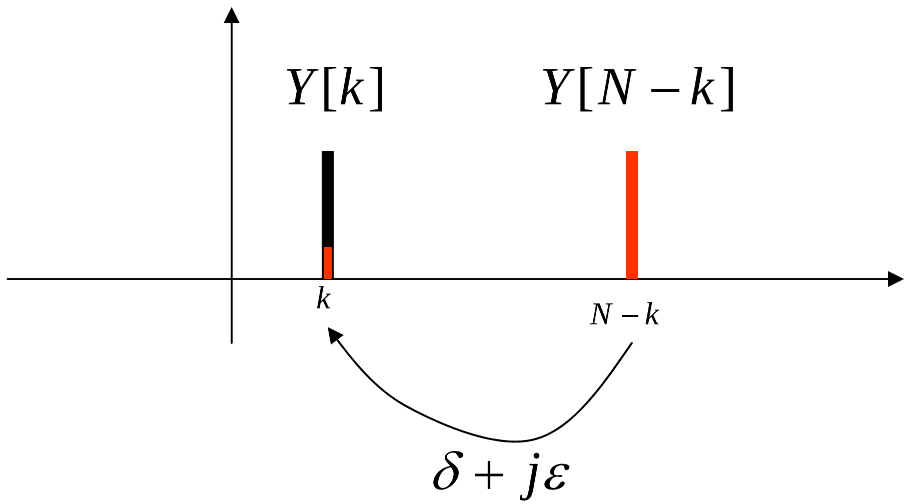
$$\overline{y}[n] = \overline{A}e^{j\overline{\varphi}}\left(y[n]+\left(\delta+j\varepsilon\right)y^*[n]\right)$$

Substitute: $$y[n] = \sum_{k=0}^{N-1} Y[k]e^{jk\frac{2\pi}{N}n}$$

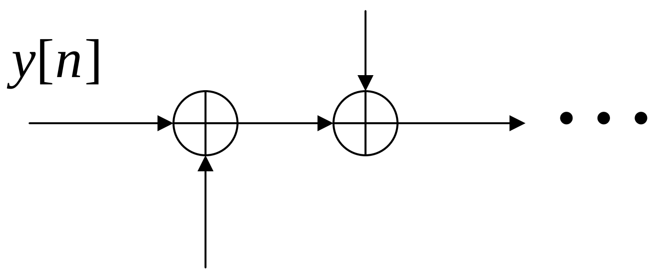
to obtain: $$\overline{y}[n] = \overline{A}e^{j\overline{\varphi}}\left(\sum_{k=0}^{N-1} Y[k]e^{jk\frac{2\pi}{N}n} + (\delta+j\varepsilon)\sum_{\ell=0}^{N-1} Y^*[\ell]e^{-j\ell\frac{2\pi}{N}n}\right)$$

$$= \overline{A}e^{j\overline{\varphi}}\left(\sum_{k=0}^{N-1}\Big(Y[k]+(\delta+j\varepsilon)Y^*[N-k]\Big)e^{jk\frac{2\pi}{N}n}\right)$$

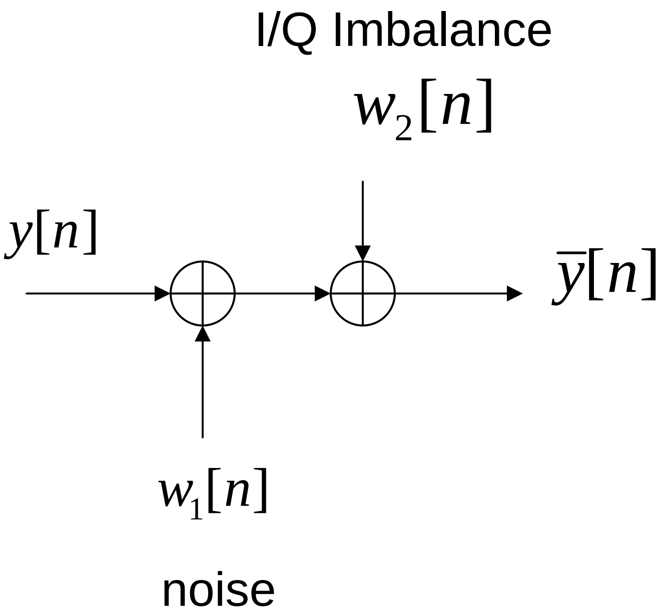
The effect is subcarrier *N-k* leaking into subcarrier *k*



It is as having extra noise added to the signal:



The factor $|\delta + j\varepsilon|$ is called Negative Frequency Rejection and it can be modeled as covariance of additive noise.



Define:

$$SNR_i = 10\log\frac{\sigma_y^2}{\sigma_i^2}$$

Implementation Loss:

$$10\log\frac{\sigma_y^2}{\sigma_1^2+\sigma_2^2} - 10\log\frac{\sigma_y^2}{\sigma_1^2} = -10\log\left(1+\frac{\sigma_2^2}{\sigma_1^2}\right) = -10\log\left(1+10^{\frac{SNR_1-SNR_2}{10}}\right)$$

Example: let

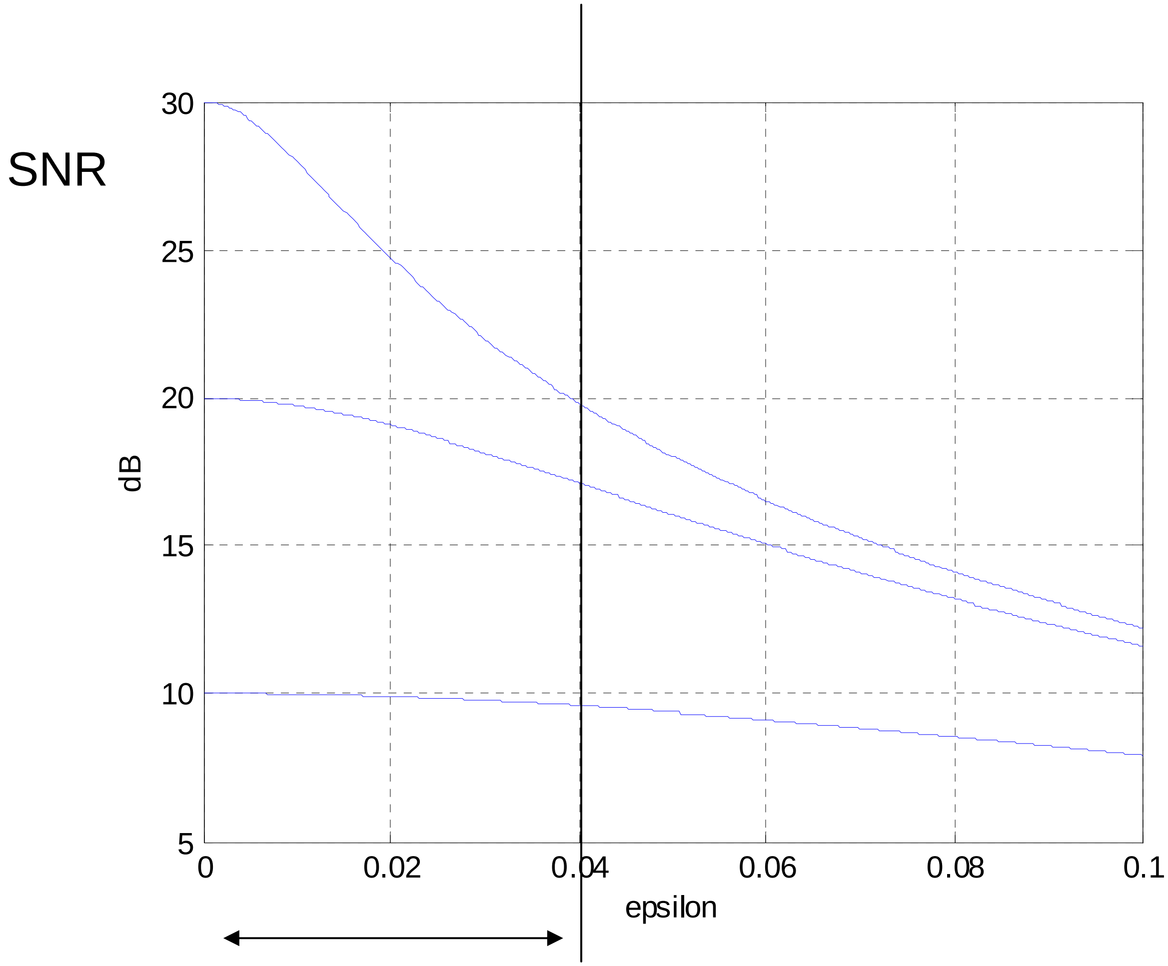
$$SNR_{noise} = 20dB$$
$$SNR_{IQ} = 35dB$$

Implementation Loss $$-10\log\left(1+10^{\frac{20-35}{10}}\right) = -0.135$$

Total SNR: $$SNR_{noise} + IL = 20 - 0.135 = 18.65dB$$

Frequency Synchronization

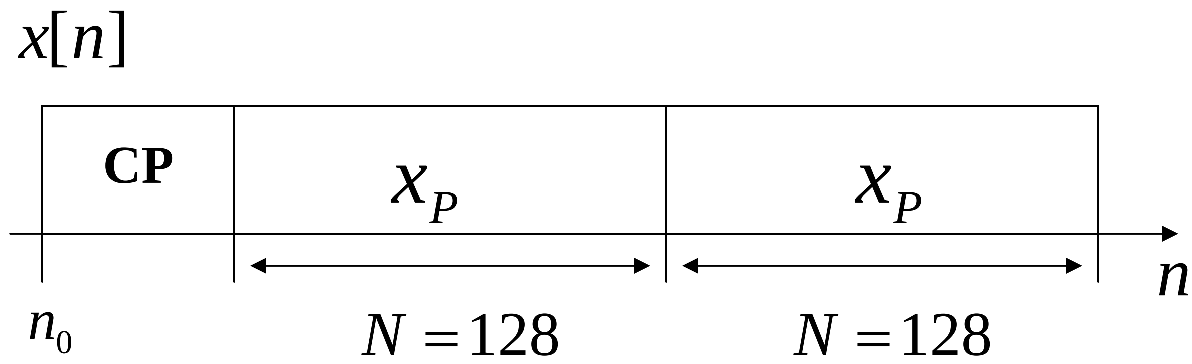
- Drifts and errors in the oscillator frequencies cause a frequency offset, like a doppler shift;
- This causes loss of subcarriers orthogonality and Inter Carrier Interference (ICI).
- The effect of ICI can be modeled as additive noise, which can be quantified



Fact: to maintain satisfactory SNR we need to limit the frequency offset to

$$\varepsilon = \frac{|\Delta F|}{F_S / N} \leq 4\%$$

• <u>Frequency Synchronization</u> from Long Preamble.



If there is a frequency offset (due to oscillator drifts or doppler shift) the received signal has an extra phase term

$$y[n] = y_0[n] e^{j 2\pi \Delta f\, n}$$

with Δf the digital frequency offset. Take the two received parts of the long preamble:

$$y_1[n] = y_0[n] e^{j(2\pi \Delta f\, n + \alpha)} + w_1[n]$$

$$y_2[n] = y_0[n] e^{j(2\pi \Delta f\, (n+N) + \alpha)} + w_2[n] \qquad n = 0, \ldots, N-1$$

Comparing the two in vector form:

$$y_2 = y_1 e^{j2\pi\Delta f\, N} + w$$

where $y_i = \begin{bmatrix} y_i[0] & \cdots & y_i[N-1] \end{bmatrix}^T$

Then:

$$y_1^{*T} y_2 = \| y_1 \|^2 e^{j2\pi\Delta f\, N} + y_1^{*T} w$$

which yields the estimate of the frequency offset

$$\Delta f = \frac{1}{2\pi N} \arctan\left(\frac{\mathrm{Im}\left\{ y_1^{*T} y_2 \right\}}{\mathrm{Re}\left\{ y_1^{*T} y_2 \right\}} \right)$$

Since, $|\arctan| < \pi$ then the frequency offset has to be at the most $|\Delta f| < \dfrac{1}{2N}$

with *N* being the FFT length.